A roadmap for statistical genetics

Taotao Tan

2024-07-26

A roadmap for statistical genetics

The field of statistical genetics involves lots of mathematical derivations. However, the derivations can be forgotten unless it is well-documented. This is my personal documentation with essential results in stats gen.

- Topic 1: Linear regression with scaled genotype and phenotype.
- Topic 2: LD score regression.
- Topic 3: Polygenic score.
- Topic 4: TWAS.

Topic 1: Linear regression with scaled genotype and phenotype

Statistical geneticist often scale genotype and phenotype before GWAS analysis. This procedure can typically simplify the mathematics, and has a few important consequences. They also tend to parameterize the model with heritability. Let's define some notations that we will use for the entire documentation.

$$
\begin{aligned}
G &: \text{standardized genotype matrix} \\
y &: \text{standardized phenotype} \\
\lambda, \hat{\lambda} &: \text{true and estimated causal effect sizes} \\
\beta, \hat{\beta} &: \text{true and estimated marginal effect sizes} \\
R &: \text{LD matrix/ correlation matrix} \\
h^2 &: \text{narrow sense heritability} \\
N &: \text{sample size} \\
M &: \text{number of variants}
\end{aligned}
$$

The model we are considering is

$$
\begin{aligned}
y &= G\lambda + \varepsilon \\
\varepsilon &\sim N(0, (1-h^2)I)
\end{aligned}
$$

Assuming each variant only explains a tiny bit of phenotypical variation, I present a few important results:

1. GWAS effect size estimates is $\hat{\beta} = \frac{1}{N} G^T y$.
2. GWAS standard error is a constant $s.e. = 1/\sqrt{N}$.
3. GWAS z score is $\hat{\beta}\sqrt{N}$.
4. LD is $R = \frac{G^T G}{N}$. it is not always invertable, but we can add a small diagonal matrix as for regularization.
5. multiple regression results is $\hat{\lambda} = R^{-1}\hat{\beta}$.
6. The underlying genetic value is $G\lambda$, therefore the heritability is defined as $h^2 = \frac{\lambda^T G^T G \lambda}{N} = \lambda^T R \lambda = \beta^T R^{-1} \beta$
7. $\hat{\lambda}$ is an unbiased estimator of λ. More specifically, $\hat{\lambda} \sim MVN(\lambda, \frac{1-h^2}{N} R^{-1})$
8. $\hat{\beta}$ is an unbiased estimator of β, More specifically, $\hat{\beta} = R\hat{\lambda} \sim MVN(R\lambda, \frac{1-h^2}{N} R)$. The diagonal elements are squared standard error of individual marker, which is usually close enough to $1/N$ when heritability is small

Additionally, due to the sample size is typically large in GWAS, any additional degree of freedom will be ignored. Therefore, it is uncommon to see d.o.f adjustment like $n - k$ in GWAS. In other words, we are performing asymptotic inference in GWAS.

Surprisingly, covariates are often ignored in the formulation: $y = G\lambda + \varepsilon$. Some model either regress out covariates, or assume $\mathbb{C}ov[G, \varepsilon] \neq 0$. Therefore, it's important to think carefully about covariates.

Here is a demonstration using simulated data:

```r
path = "https://www.mv.helsinki.fi/home/mjxpirin/GWAS_course/material/APOE_1000G_FIN_74SNPS."
haps = read.table(paste0(path,"txt"))
info = read.table(paste0(path,"legend.txt"),header = T, as.is = T)

n = 1000
G = as.matrix(haps[sample(1:nrow(haps), size = n, repl = T),] +
                haps[sample(1:nrow(haps), size = n, repl = T),])[,1:12]
row.names(G) = NULL
colnames(G) = NULL

G_ = scale(G)

# simulate 3 causal SNPs
lambda = rep(0, ncol(G_))
lambda[c(3, 4, 7)] =  c(0.3, -0.1, 0.15)
R = t(G_) %*% G_ /n # LD matrix
h2 = t(lambda) %*% R %*% lambda # heritability

# there are some randomness about err, re-scale it to have a better control of h2
err = scale(rnorm(n))
err_ = (err - mean(err)) * c(sqrt(1 - h2))

y = G_ %*% lambda + err_
y_ = scale(y)

### Key results:

beta_hat = 1/n * t(G_) %*% y_  # GWAS effect size estimates
z = sqrt(n) * beta_hat # z-score estimates
se_method1 = 1/sqrt(n) # standard error
pval = pchisq(z^2, df = 1, lower.tail = F) # p value
beta = R %*% lambda

# another way to compute standard error, which is pretty close to 1/sqrt(n)
se_method2 = sqrt(diag(as.numeric((1 - h2)/n) * R))
```

Observation: In reality, we don't know which SNP is causal, and we don't know their magnitude. It's possible to use the estimated marginal effect size to estimate the heritability with $\hat{h}^2 = \hat{\beta}^T R^{-1} \hat{\beta}$. But this approach relies on inverting the LD matrix, which is practically impossible. In the toy dataset, this approach would over estimate the true heritability.

Another approach is to use the variant with the most significant p value to estimate h^2. The toy dataset suggest it under estimates the heritability, perhaps because single variants doesn't carry all information of the locus.

```r
t(beta_hat) %*% solve(R) %*% beta_hat # use all variants
```

```
##             [,1]
## [1,] 0.09266536
```

```r
var(G_[,3] * beta_hat[3]) # use top variants
```

```
## [1] 0.07367161
```

Topic 2: LD score regression.

LDSC is proposed in this landmark paper, in which it described how LD affect the probability of a variant being significant. Under infinitesimal model, LDSC states $\mathbb{E}[\chi_j^2] = \frac{Nh^2}{M} l_j + 1$, where $l_j \equiv \sum_{k=1}^{M} r_{jk}^2$ is the LD score. To carry out the derivation, one must treat the effect size as random: $\lambda_j \sim N(0, \frac{h^2}{M})$.

In GWAS, the marginal effect size estimates (condition on true marginal effect size) is normally distributed: $\hat{\beta}_j|\beta_j \sim N(\beta_j, \frac{1}{N})$. Equivalently, $\hat{\beta}_j|\lambda \sim N(\sum_{k=1}^{M} r_{jk}\lambda_k, \frac{1}{N})$.

I first state some quantities that will be useful for the derivations. Those quantities should be easy to varify:

$$\begin{aligned}
\mathbb{E}[\lambda_j] &= 0 \\
\mathbb{E}[\lambda_j^2] &= \frac{h^2}{M} \\
\mathbb{E}[\hat{\beta}_j|\lambda_j] &= \sum_{k=1}^{M} r_{jk}\lambda_k \\
\mathbb{V}ar[\hat{\beta}_j|\lambda_j] &= \frac{1}{N} \\
\mathbb{E}[\hat{\beta}_j^2|\lambda_j] &= \mathbb{V}ar[\hat{\beta}_j|\lambda_j] + \mathbb{E}^2[\hat{\beta}_j|\lambda_j] = \frac{1}{N} + (\sum_{k=1}^{M} r_{jk}\lambda_k)^2
\end{aligned}$$

Before we investigate $\mathbb{E}[\chi_j^2]$, let's express $\mathbb{E}[\hat{\beta}_j^2]$:

$$\begin{aligned}
\mathbb{E}[\hat{\beta}_j^2] &= \mathbb{E}[\mathbb{E}[\hat{\beta}_j^2 \mid \lambda]] \\
&= \mathbb{E}[\frac{1}{N} + (\sum_{k=1}^{M} r_{jk}\lambda_k)^2] \\
&= \frac{1}{N} + \mathbb{E}[(\sum_{k=1}^{M} r_{jk}\lambda_k)^2] \\
&= \frac{1}{N} + \mathbb{E}[(r_{j1}\lambda_1 + r_{j2}\lambda_2 + ...)^2] \\
&= \frac{1}{N} + \mathbb{E}[\sum_{k=1}^{M}(r_{jk}\lambda_k)^2 + 2 \cdot \sum_{p \neq q} r_{jp}r_{jq}\lambda_p\lambda_q] \\
&= \frac{1}{N} + \sum_{k=1}^{M} r_{jk}^2 \cdot \frac{h^2}{M} \\
&= \frac{h^2}{M} l_j + \frac{1}{N}
\end{aligned}$$

Further,

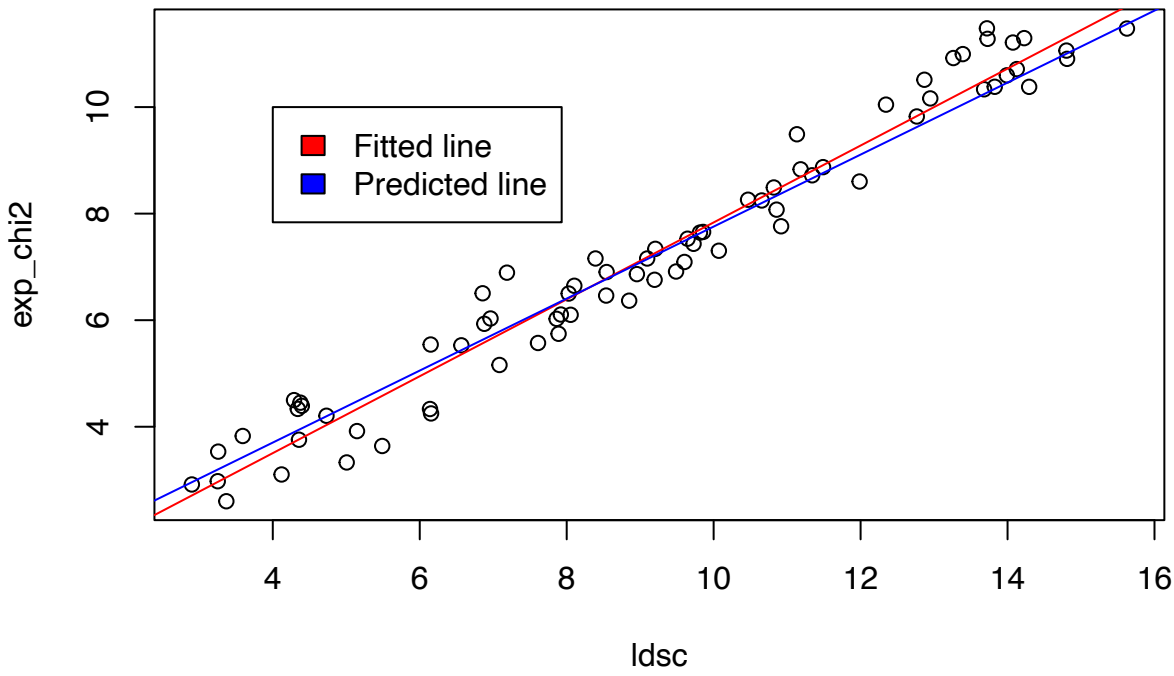
$$\begin{aligned}
\mathbb{E}[\chi_j^2] &= \mathbb{E}[(\frac{\hat{\beta}_j}{1/\sqrt{N}})^2] \\
&= N\mathbb{E}[\hat{\beta}_j^2] \\
&= \frac{Nh^2}{M} l_j + 1
\end{aligned}$$

The derivation took the insight that only marginal effect size are observed. Therefore, we investigate the statistical property of the **marginal distribution of marginal effect sizes** (a.k.a $p(\hat{\beta})$, but not the

conditional distribution $p(\hat{\beta} \mid \lambda)$). Biologically, if one variant has more LD friends, then it is more likely to be significant. LDSC has been further extended to study binary traits, partition heritability, and genetic correlation between traits.

Here is a simulation I have (with some code borrowed from Matti Pirinen's incredible tutorial).

```
path = "https://www.mv.helsinki.fi/home/mjxpirin/GWAS_course/material/APOE_1000G_FIN_74SNPS."
haps = read.table(paste0(path,"txt"))
info = read.table(paste0(path,"legend.txt"),header = T, as.is = T)

n = 1000
G = as.matrix(haps[sample(1:nrow(haps), size = n, repl = T),] + haps[sample(1:nrow(haps), size = n, repl
row.names(G) = NULL
colnames(G) = NULL

G_ = scale(G)
h2 = 0.05
R = t(G_) %*% G_ /n # LD matrix

get_chi2<- function(G_, h2, R){
  lambda = rnorm(ncol(G_), mean = 0, sd = sqrt(h2/ncol(G_)))
  err = scale(rnorm(n))
  err_ = (err - mean(err)) * c(sqrt(1 - h2))

  y = G_ %*% lambda + err_
  y_ = scale(y)

  ##
  beta_hat = t(G_) %*% y / n
  chi2 = n * beta_hat^2

  return(as.vector(chi2))
}

# do this 300 times, and average them to get the expectation
chi2_simulations = replicate(n = 300, get_chi2(G_, h2, R))
exp_chi2 = rowMeans(chi2_simulations)

ldsc = rowSums(R^2)


plot(ldsc, exp_chi2)
abline(lm(exp_chi2 ~ ldsc), col = "red")
abline(a = 1, b = (n * h2 / ncol(G)), col = "blue")
legend(4, 10, legend=c("Fitted line", "Predicted line"),
       fill = c("red","blue"))
```



Thanks to Arslan Zaidi's comment - the simulation now aligns with the derivation :)

Topic 3: Polygenic Score

Polygenic score (PRS) investigates the genetic liability of certain diseases. Given the training data, we might compute the polygenic score as $PRS_i = \sum_{j=1}^{M} \hat{\beta}_j G_{ij}$ for the testing cohort. Most of the PRS methods paper, such as PRS-CS, LDPred aim to recover causal effects λ from the observed marginal effect size estimates $\hat{\beta}_j$. Here let's consider a infinitesimal model (LDpred-inf).

We assume the causal effect size $\lambda \sim MVN(0, \frac{h^2}{M} I)$ (called infinitesimal model). From Topic 1, we also have $\hat{\beta}|\lambda \sim MVN(R\lambda, \frac{1-h^2}{N} R)$. The Bayesian inference recipe with conjugate prior normal distribution gives us (according to this document):

$$
\begin{aligned}
p(\lambda \mid \hat{\beta}) &\propto f(\hat{\beta} \mid \lambda) \cdot f(\lambda) \\
&\propto exp\{-\frac{1}{2}(\hat{\beta} - R\lambda)^T (\frac{1-h^2}{N} R)^{-1} (\hat{\beta} - R\lambda)\} \cdot exp\{-\frac{1}{2}\lambda^T (\frac{h^2}{M})^{-1} \lambda\} \\
&\propto exp\{-\frac{1}{2}[\frac{N}{1-h^2} \cdot (\hat{\beta} - R\lambda)^T R^{-1} (\hat{\beta} - R\lambda) + \frac{M}{h^2}\lambda^T \lambda]\} \\
&\propto exp\{-\frac{1}{2}[\frac{N}{1-h^2} \cdot (\hat{\beta}^T R^{-1} \hat{\beta} - \hat{\beta}^T R^{-1} R\lambda - \lambda^T R R^{-1} \hat{\beta} + \lambda^T R R^{-1} R\lambda) + \frac{M}{h^2}\lambda^T \lambda]\} \\
&\propto exp\{-\frac{1}{2}[\lambda^T (\frac{N}{1-h^2} R + \frac{M}{h^2} I)\lambda - 2\frac{N}{1-h^2}\hat{\beta}^T \lambda]\}
\end{aligned}
$$

Let $K = \frac{N}{1-h^2} R + \frac{M}{h^2} I$, $b = \frac{N}{1-h^2}\hat{\beta}$, and use the "Completing the square" technique, we have:

$$
\begin{aligned}
p(\lambda \mid \hat{\beta}) &\propto f(\hat{\beta} \mid \lambda) \cdot f(\lambda) \\
&\propto exp\{(\lambda - K^{-1}b)^T K (\lambda - K^{-1}b)\}
\end{aligned}
$$

Therefore, the posterior distribution of the causal effect size is

$$
\begin{aligned}
\lambda \mid \hat{\beta} &\sim MVN(K^{-1}b, K^{-1}) \\
&\sim MVN((R + \frac{M(1-h^2)}{Nh^2} I)^{-1}\hat{\beta}, [\frac{N}{1-h^2} R + \frac{M}{h^2} I]^{-1})
\end{aligned}
$$

One might claim that the heritability of a region is small enough, such that $1 - h^2 \approx 1$, Therefore, we can further simplify the expression, and obtain the mean and variance of the posterior causal effect size:

$$
\begin{aligned}
\mathbb{E}[\lambda \mid \hat{\beta}] &= (R + \frac{M}{Nh^2} I)^{-1}\hat{\beta} \\
\mathbb{V}ar[\lambda \mid \hat{\beta}] &= [NR + \frac{M}{h^2} I]^{-1}
\end{aligned}
$$

This expression is identical to what's mentioned in PRS-CS paper (equation 13). But I have a few more remarks about this model:

1. In both PRS-CS and LDpred manuscript, they have an additional subscript to denote a small region of the genome (in PRS-CS, LD is denoted as D_l to indicate the l-th region). This is because LD panel is pre-computed in blocks realistically.
2. This approach attempts to solve a Bayesian inference problem *without* looking at individual-level data.
3. This infinitesimal Bayesian regression approach is identical to Ridge regression.

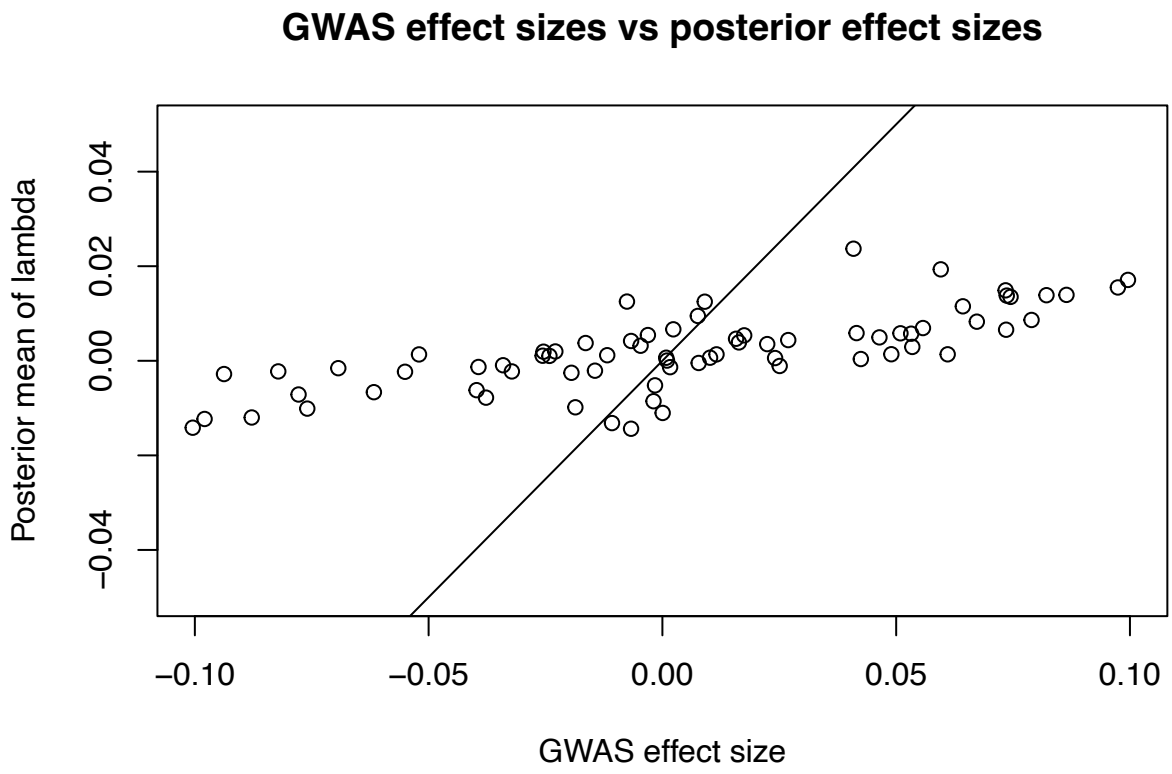
4. The heritability h^2 is treated as a parameter for the prior distribution, which must be specified according to domain knowledge before we run this analysis. This is not always trivial in realistic PRS analysis. Therefore, when we don't have any prior information about a disease, we might try grid search to find the best h^2. In machine learning lingo, this is referred as "hyper-parameter" tunning.

The framework can be further extended to multi-ancestry setting. Let's assume that we have the same causal effect sizes λ across ancestries. For each ancestry k, we have $\hat{\beta}_k|\lambda \sim MVN(R_k\lambda, \frac{1}{N}R_k)$. We might infer the posterior effect size $f(\lambda \mid \hat{\beta}_1, \hat{\beta}_2, ...)$:

$$\begin{aligned} f(\lambda \mid \hat{\beta}_1, \hat{\beta}_2, ...) &\propto f(\beta_1, \hat{\beta}_2, ... \mid \lambda) \cdot f(\lambda) \\ &\propto \prod_{k=1} f(\hat{\beta}_k \mid \lambda) \cdot f(\lambda) \end{aligned}$$

It's possible to expand the equation to find the posterior mean of $\lambda \mid \hat{\beta}_1, \hat{\beta}_2...$, which would be a function of the prior distribution of λ, observed effect sizes, and LD panel for each ancestry. However, this might not be a reasonable assumption about the prior distribution of λ, as it might be different across populations (or heritability might be different, which implies different amount of regularization). PRS-CSx instead infers the posterior effect size for each ancestry.

```
path = "https://www.mv.helsinki.fi/home/mjxpirin/GWAS_course/material/APOE_1000G_FIN_74SNPS."
haps = read.table(paste0(path,"txt"))
info = read.table(paste0(path,"legend.txt"),header = T, as.is = T)

n = 1000
G = as.matrix(haps[sample(1:nrow(haps), size = n, repl = T),] + haps[sample(1:nrow(haps), size = n, repl
row.names(G) = NULL
colnames(G) = NULL

G_ = scale(G)
h2 = 0.05
M = ncol(G_)
lambda = rnorm(ncol(G_), mean = 0, sd = sqrt(h2/M))

err = scale(rnorm(n))
err_ = (err - mean(err)) * c(sqrt(1 - h2))

y = G_ %*% lambda + err_
y_ = scale(y)

##
beta_hat = t(G_) %*% y / n
R = t(G_) %*% G_ /n # LD matrix


lambda_posterior = solve(R + M/(n *h2) * diag(M)) %*% beta_hat
plot(beta_hat, lambda_posterior, main = "GWAS effect sizes vs posterior effect sizes",
     xlim = c(-0.1, 0.1), ylim = c(-0.05, 0.05), xlab = "GWAS effect size", ylab = "Posterior mean of la
abline(a = 0, b = 1)
```



Clearly, there is a strong shrinkage of the marginal effect size.

Topic 4: TWAS

Transcriptome-wide association studies (TWAS) aims to identify associations between gene expression and trait of interest. In an ideal word where we have both RNA-seq and trait data for tens of thousands of individuals, performing a TWAS analysis would be very easy: simply regress trait by expression. However, GTEx, the largest collection of expression data, has only collected ~700 RNA-seq data without trait value. This preclude a direct association test between expression and trait. Despite the limitation, the GTEx collected the genetic data for all the participants, and trained models to predict expression value from variants across multiple tissues. The variants that strongly associate with gene expression are named as eQTLs.

For a given cohort with only genetic and phenotype data, we can first imputed/predicted RNA expression from genetics data, then regress the phenotype on the predicted RNA expressions. Let's define some additional notations:

$$\hat{x}_k : \text{standardized imputed RNA expression for gene k}$$
$$\hat{w}_k : \text{pre-trained weights from GTEx to predict gene k}$$

For a new dataset with genotype and phenotype information, we can first impute the expression by $\hat{x}_k = G\hat{w}_k$. When predicting expression level, we typically only restrict to a small region of a genome, referred as cis-eQTL. We then regress the phenotype by the predicted expression, to obtain the effect size and p-value for each gene. TWAS employs a two-stage least square regression, and is theoretically immune to any confounding between expression and trait. The model is identical to Mendelian randomization, where we treat gene expression as an exposure. With individual level genotype and phenotype information, we might perform TWAS as:

$$\begin{aligned}\hat{\beta}_k^{TWAS} &= \frac{\mathbb{C}ov[\hat{x}_k, y]}{\mathbb{V}ar[\hat{x}_k]} \\ &= \frac{\hat{x}_k^T y}{\hat{x}_k^T \hat{x}_k} \\ &= \frac{\hat{w}_k^T G^T y}{\hat{w}_k^T G^T G \hat{w}_k}\end{aligned}$$

Notice we have $\hat{\beta} = G^T y/N$, and $R = G^T G/N$, this allows us to further compress the expression to:

$$\hat{\beta}_k^{TWAS} = \frac{w_k^T \hat{\beta}}{\hat{w}_k^T R w_k} = \frac{w_k^T z}{\sqrt{N}\hat{w}_k^T R w_k}$$

As the phenotypical variance explained by a single locus is so small, the GWAS marginal effect size is distributed as $\hat{\beta} \sim MVN(R\lambda, \frac{1}{N}R)$. According to linear transformation of a multi-variate Gaussian random variable, we have:

$$\begin{aligned}\hat{\beta}_k^{TWAS} &\sim MVN(\frac{w_k^T R\lambda}{\hat{w}_k^T R w_k}, \frac{1}{N\hat{w}_k^T R w_k}) \\ s.e &= \frac{1}{\sqrt{N\hat{w}_k^T R w_k}} \\ z &= \frac{\beta_k^{TWAS}}{s.e} = \frac{\sqrt{N}\hat{w}^T\hat{\beta}}{(\hat{w}_k^T R w_k)^{\frac{1}{2}}}\end{aligned}$$

The above expression is convenient, as it allows for TWAS analysis with only GWAS summary statistics and a matched LD panel. The expression seems consistent with Sasha Gusev's presentation. I also attached some simulation to demonstrate this quantity.

```r
path = "https://www.mv.helsinki.fi/home/mjxpirin/GWAS_course/material/APOE_1000G_FIN_74SNPS."
haps = read.table(paste0(path,"txt"))
info = read.table(paste0(path,"legend.txt"),header = T, as.is = T)


### Consider this is GTEx data
n1 = 1000
G1 = as.matrix(haps[sample(1:nrow(haps), size = n1, repl = T),1:10] +
                haps[sample(1:nrow(haps), size = n1, repl = T),1:10])
row.names(G1) = NULL
colnames(G1) = NULL

G1_ = scale(G1)
M = ncol(G1_)
lambda = c(0, 0, 0, 0.3, 0, 0, 0, 0, 0, 0) # a very strong eQTL

x1 = G1_ %*% lambda + rnorm(n1, mean = 0, sd = sqrt(1 - var(G1_ %*% lambda)))
x1_ = scale(x1)
# the expression model uses all the SNPs, might need to add regularization term in realistic analysis
w = solve(t(G1_) %*% G1_) %*% t(G1_) %*% x1_

### Consider this is my own data with genotype and trait
n2 = 6000
G2 = as.matrix(haps[sample(1:nrow(haps), size = n2, repl = T),1:10] +
                haps[sample(1:nrow(haps), size = n2, repl = T),1:10])

row.names(G2) = NULL
colnames(G2) = NULL
G2_ = scale(G2)
R = cor(G2_)
beta_twas = 0.6

# x2 is not unknown, but we generate the true expression data
x2 = G2_ %*% lambda + rnorm(n2, mean = 0, sd = sqrt(1 - var(G2_ %*% lambda)))
x2_ = scale(x2)

y2 = x2_ %*% beta_twas + rnorm(n2, mean = 0, sd = sqrt(1 - var(x2_ %*% beta_twas)))
y2_ = scale(y2)

# with individual level data
beta_twas_hat_method1 = cov(y2_, G2_ %*% w)/var(G2_ %*% w)

# a GWAS effect size is pre-computed
beta_hat = 1/n2 * (t(G2_) %*% y2_)

# with sumstats level data and LD
beta_twas_hat_method2 = t(w) %*% beta_hat / (t(w) %*% R %*% w)
# z score
z_twas_method2 = sqrt(n2) *  t(w) %*% beta_hat / sqrt(t(w) %*% R %*% w)
```

A few remarks about TWAS:

1. TWAS can be interpreted as a special usage of Mendelian randomization. One might think TWAS can

identify gene expressions that causally affect the phenotype, as MR does. But due to complications such as co-expression, this is generally not true.

2. Standard TWAS analysis ignored the variability of the imputed expression data, which might induce inflated False positives. Xue et al. examnined this query, and concludes that it is mostly fine.

3. The weights for predicting gene expression are often obtained from GTEx, but would these weights as predictive across different ancestries? Chen et al. presented a method that integrate multiple GWAS results to perform TWAS. But I think there are rooms for more methodology development.