Tutorial: statistical fine-mapping in genetics association studies

Gao Wang

rm(list=ls())  
library(susieR)  
data(N3finemapping)  
attach(N3finemapping)

y = Y[,1]  
b = true\_coef[,1]

**Question: Please use R function plot to visualize the true effects vector. What are the effect sizes for these 3 "causal" variants?**

**Answer:**

plot(b, pch=16, ylab='effect size')

![](data:image/png;base64;base64,)

The effect size of the 3 effect variants are:

b[which(b != 0)]

1. 1.18917229920409
2. -0.831788980799581
3. 1.39928773430517

sumstats <- univariate\_regression(X, y)

**Question: Please use R function plot to compare the true effects variable b with the estimated effect sizes sumstats$betahat. Are they consistent, and why?**

**Answer:**

The estimated effect sizes (sumstats$betahat) are:

plot(sumstats$betahat, ylab='effect size',pch=20)

![](data:image/png;base64;base64,)

We add the true effects b in red to compare with the estimated effect sizes sumstats$betahat in black:

plot(sumstats$betahat, ylab='effect size',pch=20)+  
points(b,col=2,pch=16,cex=0.8)

The estimated effects via univariate regression are very different from the simulated true effects. Many non-effect variables show large effects (and z-scores, see below) due to inflation induced by linkage disequilibrium (LD) with the effect variables.

plot(sumstats$betahat/sumstats$sebetahat, ylab='z-scores',pch=20)

![](data:image/png;base64;base64,)

Below is visualization of correlations between variables near the first effect variable, showing high correlations between variants in genetic data:

corrplot::corrplot(cor(X[,380:430]))

![](data:image/png;base64;base64,)

z\_scores <- sumstats$betahat / sumstats$sebetahat  
log10p <- -log10(pchisq(z\_scores^2,1,lower.tail=F))

**Question: How many variables have p-values smaller than ? Would you report all of them as significant association findings, and if not, which variable(s) will you likely report, based on univariate association analysis?**

**Answer:**

gwas\_sig = which(log10p>8)  
cat (length(gwas\_sig),'variables have p-values smaller than 10^-8.')

5 variables have p-values smaller than 10^-8.

Correlation between these variables are very high:

gwas\_sig

1. 653
2. 773
3. 776
4. 777
5. 794

corrplot::corrplot(cor(X[,gwas\_sig]))

![](data:image/png;base64;base64,)

Given the high correlations among these variables, it is very likely that only 1 out of the 5 variables is the true effect variable. Without performing formal analysis to account for LD, the variable with smallest p-value from this cluster of highly correlated variables are typically reported as the "causal" variable.

lliks = sapply(1:ncol(X), function(i) logLik(lm(y~X[,i])))

lliks = lliks - max(lliks) # To avoid taking exp on large numbers in the next line, thus improving numerical stability   
probs = exp(lliks)/sum(exp(lliks))

which(susieR:::in\_CS(t(probs), coverage = 0.95) != 0)

1. 773
2. 777

**Question: Show the p-values and probabilities of non-zero effect for these two variables. Compute the correlation between these two variables, and commment on result for the zero effect variable 777**.

**Answer**

cat("−𝑙𝑜𝑔10 p-values from above:")  
log10p[c(773,777)]  
cat("p-values:")  
10^-(log10p[c(773,777)])

−𝑙𝑜𝑔10 p-values from above:

cat("'causal' probability:")  
probs[c(773,777)]

'causal' probability:

Correlation

cor(X[,773],X[,777])

0.981572576486809

The zero-effect variable 777 shows a small p-value because it is highly correlated (0.98 Pearson's correlation) with the effect variable 773. However, under the assumption that only one variable is non-zero effect, the probability of 773 being the variable is much higher than that for 777. In this example, statistical fine-mapping performs quite well even though correlations between variables are high.

fitted <- susie(X, y, L = 10)

sets <- susie\_get\_cs(fitted, X = X, coverage = 0.9)

**Question: why is the size for the 3rd CS smaller and minimum absolute correlation higher, when requesting 90% coverage compared to previous result of requesting 95%?**

**Answer:**

By definition of CS, less variables are needed to sum up to 90% compared to that needed to sum up to 95%. Because less variables are involved, the size of CS are smaller and the minimum correlation between variables are higher.

**Question: Report the PIP for variable 403. Why is the PIP relatively low for this true effect variable?** You can use the code below to compare p-value and PIP for the CS containing variable 403:

i <- fitted$sets$cs$L3  
z3 <- cbind(i,log10p[i],fitted$pip[i])  
colnames(z3) <- c('position', '-log10 pvalue', 'PIP')

**Answer:**

z3[z3[,1]==403,'PIP']

**PIP:** 0.0319928476988085

sum(fitted$pip[i])

0.953981494965999

length(i)

27

min(cor(X[,i]))

0.868630914830401

Because 27 variants are captured in the CS and these variables are highly correlated (at least 0.86 in Pearson's correlation), the probability of any given variable being the effect variable is small, even though collectively the probability of any of them has non-zero effect is high (greater than 0.95).

**Question: Please rerun the susie analysis setting L=1 to detect a maximum of only one effect across the region. How does the result compare to the previous section of "one causal effect" fine-mapping using likelihoods from univariate analysis?**

**Answer:**

fitted\_1 <- susie(X, y, L = 1)  
susie\_plot(fitted\_1, y="PIP", b=b)

![](data:image/png;base64;base64,)

The result here is similar with the previous section of "one causal effect" fine-mapping using likelihoods from univariate analysis, although here Bayesian regression analysis were performed instead of linear regression implemented in lm() in R.

**Question: Please change scaled\_prior\_variance to different numbers between 0 and 1, and examine the results. What happens if you set it to 0.001, and why?**

**Answer:**

par(mfrow=c(1,5))   
  
fitted\_3 = susie(X, y, L = 10, estimate\_prior\_variance = FALSE, scaled\_prior\_variance = 0.01)  
susie\_plot(fitted\_3, y='PIP', b=b)  
  
fitted\_4 = susie(X, y, L = 10, estimate\_prior\_variance = FALSE, scaled\_prior\_variance = 0.02)  
susie\_plot(fitted\_4, y='PIP', b=b)  
  
fitted\_5 = susie(X, y, L = 10, estimate\_prior\_variance = FALSE, scaled\_prior\_variance = 0.03)  
susie\_plot(fitted\_5, y='PIP', b=b)  
  
fitted\_6 = susie(X, y, L = 10, estimate\_prior\_variance = FALSE, scaled\_prior\_variance = 0.1)  
susie\_plot(fitted\_6, y='PIP', b=b)  
  
fitted\_7 = susie(X, y, L = 10, estimate\_prior\_variance = FALSE, scaled\_prior\_variance = 0.8)  
susie\_plot(fitted\_7, y='PIP', b=b)

![](data:image/png;base64;base64,)

The results are robust to the choice of prior effect size, suggesting that there is strong enough signal in data and the results are mostly driven by data rather than by prior effect size specified.

Very small prior effect reflects the belief that a variable should only contribute to a very small proportion of variation in response. For example if we assume a variant should explain only 0.1% of heritability,

fitted\_7 = susie(X, y, L = 10, estimate\_prior\_variance = FALSE, scaled\_prior\_variance = 0.001)  
susie\_plot(fitted\_7, y='PIP', b=b, add\_legend=T)

![](data:image/png;base64;base64,)

The true effects are still identified, but the PIP is low because the very small prior nudges the posterior estimates towards zero.

R <- cor(X)

fitted\_rss <- susie\_rss(z\_scores, R, L = 10)

set.seed(1234)  
tmp = matrix(rnorm(500\*1001), 500, 1001)  
eigenR = eigen(R)  
eigenR$values[eigenR$values < 1e-10] = 0  
X\_ref = tmp %\*% (eigenR$values \* t(eigenR$vectors))  
R\_ref = cor(X\_ref)

fitted\_rss\_ref <- susie\_rss(z\_scores, R\_ref, L = 10)

n\_ref = 500  
fitted\_rss\_ref\_corrected <- susie\_rss(z\_scores, R\_ref, z\_ld\_weight = 1/n\_ref, L = 10)

**Question: Please compare and comment on the results using adjusted reference LD panel. Is the result improved?**

**Answer:**

The adjusted LD reference resulted in less false discovers compared to the uncorrected reference, although it still failed to capture all effect variables compared to using the correct LD reference.

data(N2finemapping)

Please use the same R code for N3finemapping data to analyze this new data-set, similarly focusing on the first column of the phenotype matrix as the phenotype of interest. This data-set contains two true effect variables. However, as you are soon to find out in univariate linear regression analysis, **the lead SNP via univariate analysis is neither of the two effect variables**.

**Question: Perform fine-mapping under "one causal variant" assumption. Does it capture any true effect? Does fine-mapping with SuSiE capture all true effects?**

**Answers:**

dim(N2finemapping$X)  
dim(N2finemapping$Y)  
y\_2 = N2finemapping$Y[,1]  
b\_2 = N2finemapping$true\_coef[,1]  
which(b\_2 != 0)

1. 574
2. 1002

sumstats\_2 <- univariate\_regression(N2finemapping$X, y\_2)  
z\_scores\_2 <- sumstats\_2$betahat / sumstats\_2$sebetahat  
log10p\_2 <- -log10(pchisq(z\_scores\_2^2,1,lower.tail=F))

susie\_plot(z\_scores\_2,y="z",b=b\_2)

![](data:image/png;base64;base64,)

**One causal assumption:**

lliks\_2 = sapply(1:ncol(N2finemapping$X), function(i) logLik(lm(y\_2~N2finemapping$X[,i])))  
lliks\_2 = lliks\_2 - max(lliks\_2) # To avoid taking exp on large numbers in the next line, thus improving numerical stability   
probs\_2 = exp(lliks\_2)/sum(exp(lliks\_2))

susie\_plot(probs\_2, y="PIP", b=b\_2, ylab = "Prob. true effect (assuming one effect variable)")

![](data:image/png;base64;base64,)

As expected, under the "one causal variant" assumption, neither of the true effect variables in red is identified.

**SuSiE Fine mapping:**

**95% credible sets:**

fitted\_N2 <- susie(N2finemapping$X, y\_2, L = 2)  
print(fitted\_N2$sets)

$cs  
$cs$L2  
 [1] 850 913 914 915 916 920 924 925 926 927 930 931 933 934 935  
[16] 942 946 948 951 952 962 967 968 979 980 982 983 985 988 989  
[31] 993 994 996 999 1000 1001 1002  
  
$cs$L1  
[1] 337 379 440  
  
  
$purity  
 min.abs.corr mean.abs.corr median.abs.corr  
L2 0.9722386 0.9939738 0.9949722  
L1 0.8534981 0.9183993 0.8944609  
  
$cs\_index  
[1] 2 1  
  
$coverage  
[1] 0.9545953 0.9696343  
  
$requested\_coverage  
[1] 0.95

**90% credible sets:**

sets\_2 <- susie\_get\_cs(fitted\_N2, X = N2finemapping$X, coverage = 0.9)  
print(sets\_2)

$cs  
$cs$L1  
[1] 337  
  
$cs$L2  
 [1] 913 914 915 916 920 925 926 927 930 931 933 934 935 942 948  
[16] 951 952 962 967 968 979 985 988 993 994 996 999 1000 1001 1002  
  
  
$purity  
 min.abs.corr mean.abs.corr median.abs.corr  
L1 1.0000000 1.0000000 1.0000000  
L2 0.9780338 0.9948644 0.9954347  
  
$cs\_index  
[1] 1 2  
  
$coverage  
[1] 0.912284 0.900496  
  
$requested\_coverage  
[1] 0.9

susie\_plot(fitted\_N2, y="PIP", b=b\_2)

![](data:image/png;base64;base64,)

Fine-mapping assuming more than one effect captures both simulated true effects.