Non additive interation

Ziang Zhang

3/25/2021

Contents

1 The Underlying Model:

In the first simulation study, we consider the following additive probit model with non additive interaction:

$$Y^* = \beta_0 + \beta_G G + \beta_E E + \gamma_1 I(G=1) * E + \gamma_2 I(G=2) * E + \epsilon$$

where $\epsilon \sim N(0, \sigma^2)$ for some **known** σ^2.

This model is more general than the model with additive interaction effect. When the interaction effect between G and E is actually additive, we should have $2\gamma_1 = \gamma_2$. Here our main interest will be testing the null hypothesis

$$H_0 : \gamma_1 = \gamma_2 = 0$$

without the information of E.

Similar to the previous situation, the presence of term γ_1, γ_2 breaks the homoskedasiticiy assumption on $Var(Y^*|G)$, and hence result in a genotypic probit model (instead of the additive model)

$$Y^* = \tilde{\gamma_0} + \tilde{\gamma_1} I(G=1) + \tilde{\gamma_2} I(G=2) + \tilde{\epsilon}$$

We will utilize the non-linearity test to test $H_0 : 2\tilde{\gamma_1} = \tilde{\gamma_2}$.

2 Simulation :

2.1 An example when the interaction effect is additive:

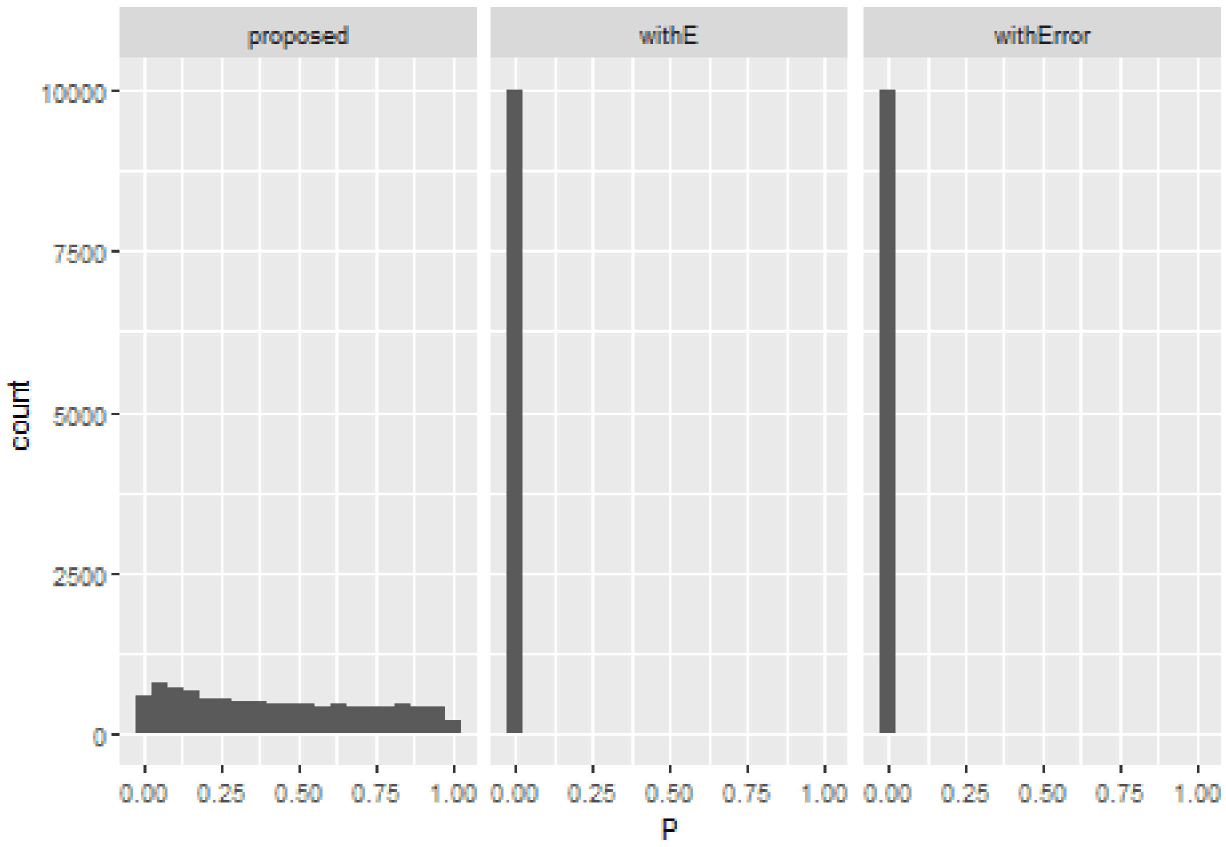
```
### Read in data:
path <- "D:/gwas-practice/indep_QC.bed"
tmpfile  <- tempfile()
snp_readBed(path, backingfile = tmpfile)
```

```
## [1] "C:\\Users\\aguer\\AppData\\Local\\Temp\\RtmpELe8GK\\file3b042503552b.rds"
```

```
obj.bigSNP <- snp_attach(paste0(tmpfile , ".rds"))

G   <- obj.bigSNP$genotypes
CHR <- obj.bigSNP$map$chromosome
POS <- obj.bigSNP$map$physical.pos

### Randomly Sample p genes
set.seed(123,sample.kind="Rounding")
p <- 5
### Need to make sure that all genotypes have enough frequencies in the selected genes:
freq_counts <- big_counts(G)
MAF <- snp_MAF(G)
Qualified <- freq_counts[3,] >= 200 & MAF>=0.3
indx <- which(POS %in% sample(POS[Qualified], size = p))
G_use <- G[,indx]
POS_use <- POS[indx]
CHR_use <- CHR[indx]
### Do the bootstrapping:
bootsize <- 5000
G_boot <- G_use[sample(1:nrow(G_use), bootsize, replace=TRUE), ]

### Specify a set of parameter, Compute power:
b0 <- -1
bG <- 0.3
gam1 <- 0.3
gam2 <- 0.6
bE <- 0.3
sig <- 1
sigmaE <- 1
measure_error_percentage <- 1/4
power <- Interaction_Test(G = G_boot[,1], b0, bG, gam1, gam2, bE, sig, sigmaE, K = 10000, measure_error
apply(as.matrix(power)<=0.05, 2, mean)
```

```
##  proposed     withE withError
##     0.099     1.000     1.000
```

```
power %>% pivot_longer(proposed:withError, values_to = "P", names_to = "type") %>% ggplot(aes(P)) + geom
```



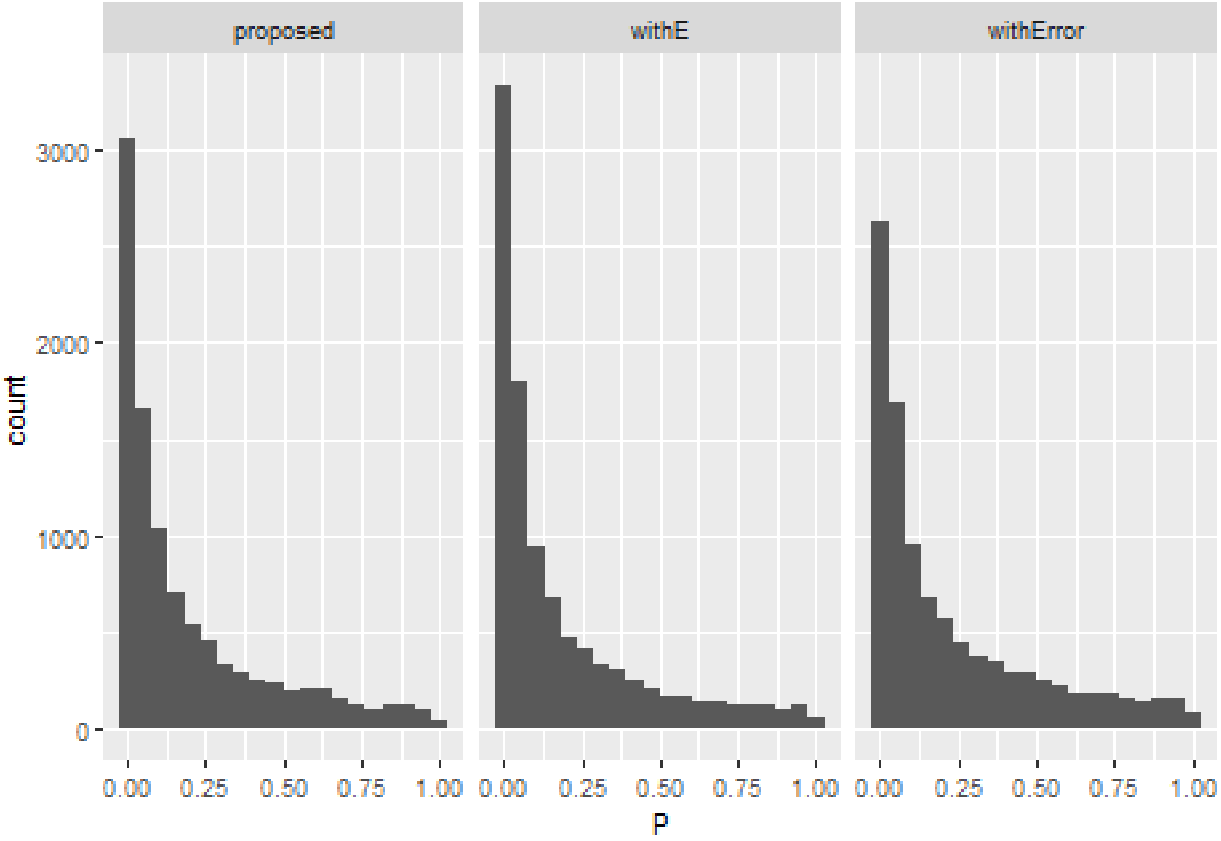
2.2 An example when the interaction effect is non-additive:

```
### Randomly Sample p genes
set.seed(123,sample.kind="Rounding")
p <- 5
### Need to make sure that all genotypes have enough frequencies in the selected genes:
freq_counts <- big_counts(G)
MAF <- snp_MAF(G)
Qualified <- freq_counts[3,] >= 200 & MAF>=0.3
indx <- which(POS %in% sample(POS[Qualified], size = p))
G_use <- G[,indx]
POS_use <- POS[indx]
CHR_use <- CHR[indx]
### Do the bootstrapping:
bootsize <- 5000
G_boot <- G_use[sample(1:nrow(G_use), bootsize, replace=TRUE), ]

### Specify a set of parameter, Compute power:
b0 <- -1
bG <- 0.3
gam1 <- 0.3
gam2 <- 0
bE <- 0.3
sig <- 1
```

```
sigmaE <- 1
measure_error_percentage <- 1/4
power <- Interaction_Test(G = G_boot[,1], b0, bG, gam1, gam2, bE, sig, sigmaE, K = 10000, measure_error_
apply(as.matrix(power)<=0.05, 2, mean)
```

```
##  proposed     withE withError
##    0.3978    0.4315    0.3536
```

```
power %>% pivot_longer(proposed:withError, values_to = "P", names_to = "type") %>% ggplot(aes(P)) + geom
```



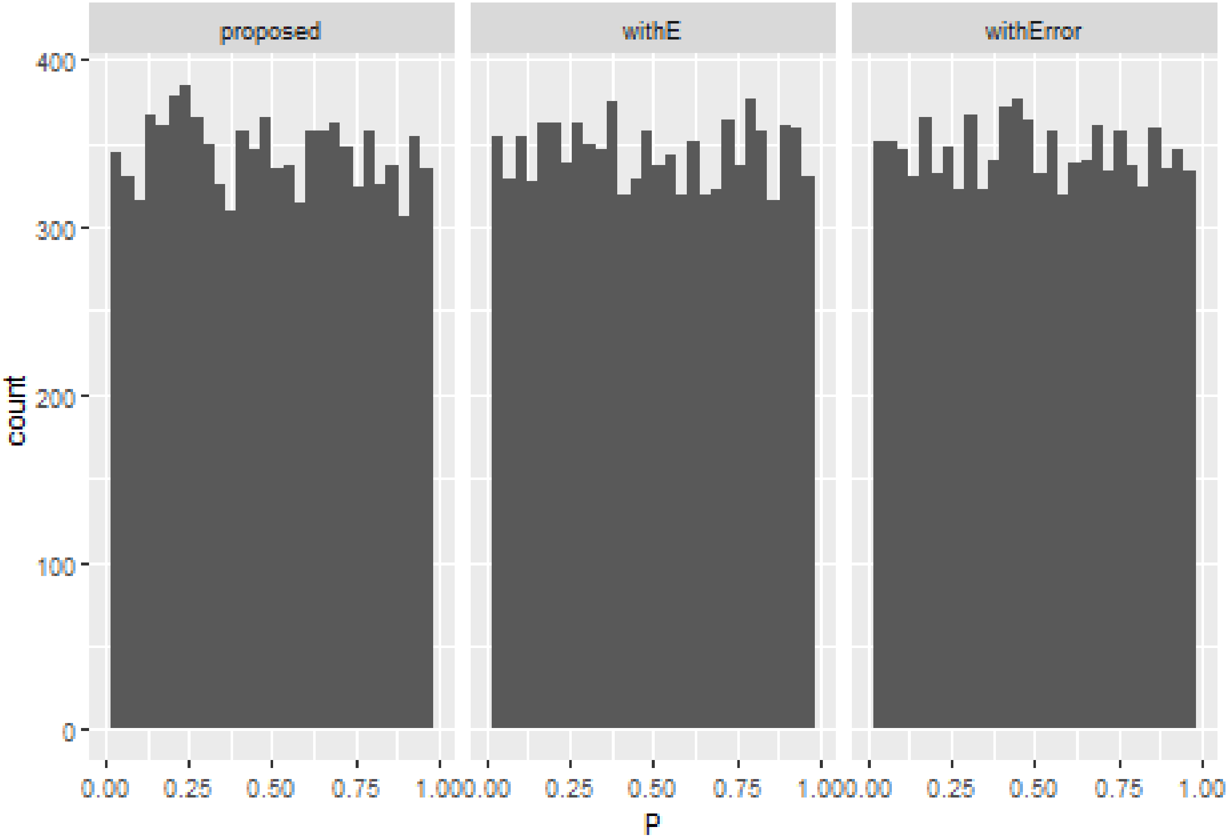
2.3 Study of Type I error rate:

```
set.seed(123,sample.kind="Rounding")
bootsize <- 5000
G_boot <- G_use[sample(1:nrow(G_use), bootsize, replace=TRUE), ]

### Specify a set of parameter, compute Type I error rate
b0 <- -1
bG <- 0.3
gam1 <- 0
gam2 <- 0
bE <- 0.3
```

```
sig <- 1
sigmaE <- 1
measure_error_percentage <- 1/4
error <- Interaction_Test(G = G_boot[,1], b0, bG, gam1, gam2, bE, sig, sigmaE, K = 10000, measure_error_
apply(as.matrix(error)<=0.05, 2, mean)
```

```
##  proposed     withE withError
##    0.0493    0.0483    0.0504
```

```
error %>% pivot_longer(proposed:withError, values_to = "P", names_to = "type") %>% ggplot(aes(P)) + geom
```


proposed
withE
withError
count
P