Simulations about coining

Georgia Salanti

2 February 2018

The data from the article Evangelou et al. PLoS Genet. 2006

Load the dataset dataS1.csv:

```
library(readxl)
library(meta)

dataS1 <- read.csv("~/_mydrive/R projects/Simulations_coining/dataS1.csv")
head(dataS1)
```

```
##   X      logORU  selogORU     logORF  selogORF  pooledOR
## 1 1 -0.01121245 0.2734163  0.8303483 0.2906279 1.4680162
## 2 2 -0.44289619 0.2844788 -1.6094379 0.7745967 0.5590371
## 3 3  0.00349142 0.2119904  0.9886114 0.2928409 1.4080791
## 4 4 -0.45445122 0.2053071 -1.3862944 0.7905694 0.5985019
## 5 5  0.33370215 0.3176995 -0.4353181 0.3869530 1.0243401
## 6 6 -0.04215027 0.1031681  0.5705449 0.3469890 1.0076428
```

Note that the average standard errors are average smaller in the case-control studies

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.06463 0.13193 0.17756 0.18846 0.21251 0.49826

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0687  0.1775  0.2278  0.2628  0.2912  0.7906
```

Simulations of meta-analysis 'true' treatment effects: Bias in the underlying true RORs

We first run simulations targeting the true underlying effects ignoring sampling errors. We assume that the association between the exposure and the outcome in subgroup 1 of a meta-analysis is reflected in OR1; similarily for subgroup 2 in the same meta-analysis (OR2).

Assuming zero subgroup ORs, no subgroup difference, same heterogeneity for each subgroup

We generate the “true” underlying ORs for the two subgroups (OR1 and OR2) for 100 000 meta-analyses. The expected means are 1 for each subgroup ans the heterogeneity SD is 0.25 (the average observed within-subgoup heterogeneity in the 92 meta-analysis). We expect that the ROR will also be one. As simulations show below, there is no bias produced with coining.

```
logOR1=rnorm(100000,log(1),0.25)
logOR2=rnorm(100000,log(1),0.25)
logRoR=logOR1-logOR2
logRoR=mean(logRoR)
cat("The ROR without coining is: ",exp(logRoR))

## The ROR without coining is:  1.000599

#coining
pooled=apply(cbind(logOR1,logOR2),1,mean)
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin
logRoRcoin=mean(invlogOR1-invlogOR2)
cat("\n The ROR after coining is: ",exp(logRoRcoin))

##
##  The ROR after coining is:  0.9998863
```

Assuming zero subgroup ORs, no subgroup difference, different heterogeneities per subgroup

Let us now assume that the heterogeneity is not the same in each subgroup; SD heterogeneity is 0.25 in one group and 0.50 in the family-based studies. Then bias is introduced and the mean ROR with coining is 0.77 instead of 1.

```
logOR1=rnorm(100000,log(1),0.25)
logOR2=rnorm(100000,log(1),0.50)
logRoR=logOR1-logOR2
logRoR=mean(logRoR)
cat("The ROR without coining is: ",exp(logRoR))

## The ROR without coining is:  0.9987658

#coining
pooled=apply(cbind(logOR1,logOR2),1,mean)
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin
logRoRcoin=mean(invlogOR1-invlogOR2)
cat("\n The ROR after coining is: ",exp(logRoRcoin))
```

```
##
##  The ROR after coining is:  0.7652965
```

Assuming subgroup differences and same heterogeneity for each subgroup

Let us now assume the most likeliy scenario: that the heterogeneity is different in each subgroup and the ORs in each subgroup are different so that the exposure relates to the outcome and the expected ROR is not 1. We assume OR1=1.8 and OR2=1.2 so that the expected ROR is 1.5. Then "coining" gives a ROR = 1.26.

```
logOR1=rnorm(100000,log(1.8),0.25)
logOR2=rnorm(100000,log(1.2),0.50)
logRoR=logOR1-logOR2
logRoR=mean(logRoR)
cat("The ROR without coining is: ",exp(logRoR))

## The ROR without coining is:  1.502047

#coining
pooled=apply(cbind(logOR1,logOR2),1,mean)
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin
logRoRcoin=mean(invlogOR1-invlogOR2)
cat("\n The ROR after coining is: ",exp(logRoRcoin))

##
##  The ROR after coining is:  1.263454
```

Simulations of study data: bias in the estimated ROR

We run the same simulations as above, but now 10 000 are generated and we attach random-error uncertainty (bias re-sampling from the 92 standard errors for the family-based and case-control studies).

Assuming zero subgroup ORs, no subgroup difference, same heterogeneity for each subgroup

We generate the "true" underlying ORs for the two subgroups (OR1 and OR2) for 10 000 meta-analyses. The expected means are 1 for each subgroup ans the heterogeneity SD is 0.20. We expect that the ROR will also be one. As simulations show below, coining produces bias because the average SEs in one subgroup (taken from the family-based studies) is on average larger than the average SEs in the case-control studies.

```
library(meta)
logOR1=rnorm(10000,log(1),0.25)
logOR2=rnorm(10000,log(1),0.25)
```

```r
#generate SEs
selogOR1=sample(dataS1$selogORU,10000,replace=T)
selogOR2=sample(dataS1$selogORF,10000,replace=T)
pooled=metagen(TE=c(logOR1,logOR2),seTE=c(selogOR1,selogOR2),byvar=c(1:10000,
1:10000),comb.random=F)$TE.fixed.w
#coining
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin

#without coining
logRoR=logOR1-logOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoR=metagen(logRoR,selogRoR)$TE.random

#the coined
logRoRcoin=invlogOR1-invlogOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoRcoin=metagen(logRoRcoin,selogRoR)$TE.random

cat("The ROR without coining is: ",exp(pooledlogRoR))

## The ROR without coining is:  0.9963654

cat("\n The ROR after coining is: ",exp(pooledlogRoRcoin))

##
##  The ROR after coining is:  1.05268
```

Assuming zero subgroup ORs, no subgroup difference, different heterogeneities per subgroup

Let us now assume that the heterogeneity is not the same in each subgroup; SD heterogeneity is 0.20 in the case-control studies and 0.4 in the family-based studies. Then bias is introduced and the mean ROR with coining is 0.84 instead of 1.

```r
logOR1=rnorm(10000,log(1),0.25)
logOR2=rnorm(10000,log(1),0.50)
#generate SEs
selogOR1=sample(dataS1$selogORU,10000,replace=T)
selogOR2=sample(dataS1$selogORF,10000,replace=T)
#pool subgroups
pooled=metagen(TE=c(logOR1,logOR2),seTE=c(selogOR1,selogOR2),byvar=c(1:10000,
1:10000),comb.random=F)$TE.fixed.w
#coining
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin

#without coining
```

```
logRoR=logOR1-logOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoR=metagen(logRoR,selogRoR)$TE.random

#the coined
logRoRcoin=invlogOR1-invlogOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoRcoin=metagen(logRoRcoin,selogRoR)$TE.random
```

```
cat("The ROR without coining is: ",exp(pooledlogRoR))
```

```
## The ROR without coining is:  1.006784
```

```
cat("\n The ROR after coining is: ",exp(pooledlogRoRcoin))
```

```
##
##  The ROR after coining is:  0.8627764
```

Assuming subgroup differences and same heterogeneity for each subgroup

Let us now assume that the heterogeneity is the same in each subgroup but the ORs in each subgroup are different so that the exposure relates to the outcome and the expected ROR is not 1. We assume OR1=2 and OR2=1.2 so that the expected ROR is 1.65. Then "coining" gives a ROR = 1.45.

```
logOR1=rnorm(10000,log(2),0.25)
logOR2=rnorm(10000,log(1.2),0.50)
#generate SEs
selogOR1=sample(dataS1$selogORU,10000,replace=T)
selogOR2=sample(dataS1$selogORF,10000,replace=T)
#pool subgroups
pooled=metagen(TE=c(logOR1,logOR2),seTE=c(selogOR1,selogOR2),byvar=c(1:10000,
1:10000),comb.random=F)$TE.fixed.w
#coining
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin

#without coining
logRoR=logOR1-logOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoR=metagen(logRoR,selogRoR)$TE.random

#the coined
logRoRcoin=invlogOR1-invlogOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoRcoin=metagen(logRoRcoin,selogRoR)$TE.random
```

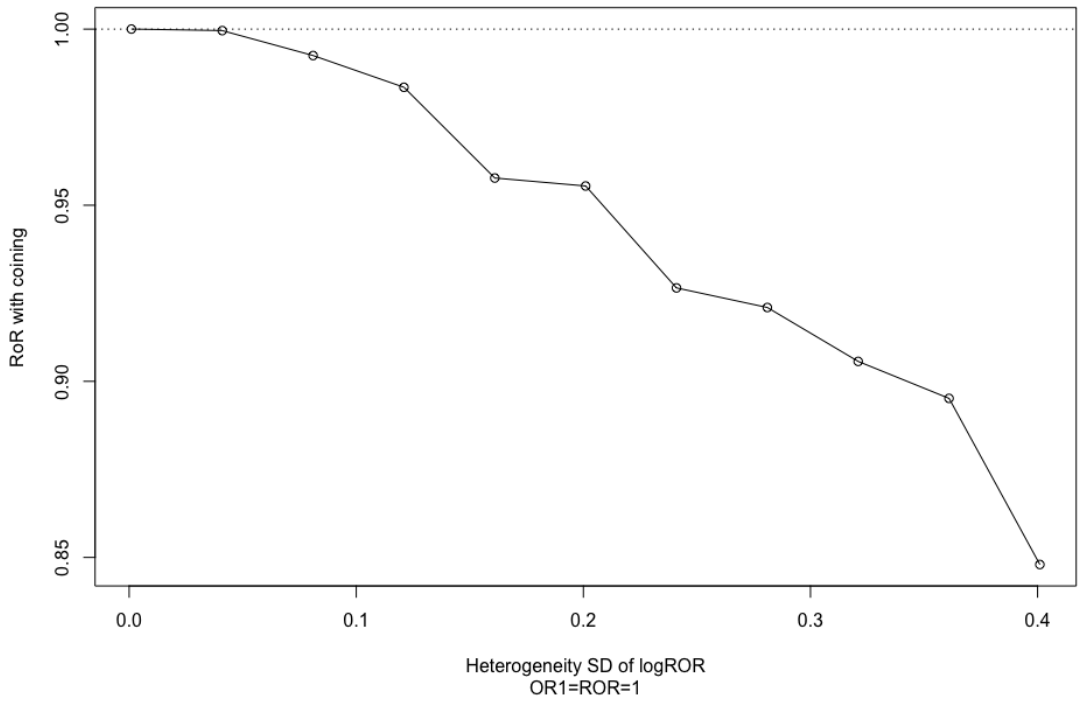
```
cat("The ROR without coining is: ",exp(pooledlogRoR))

## The ROR without coining is:  1.665005

cat("\n The ROR after coining is: ",exp(pooledlogRoRcoin))

##
##  The ROR after coining is:  1.450901
```

Dependance of bias on the heterogeneity and the true ROR

The following simulations graph summarizes the association between the heterogeneity in logROR (assumed to be ROR=1) and the bias produced by coining. To save running time only 1000 meta-analyses where assumed per heterogeneity

```
##Zero effects, no subgroup difference and differces in heteroeneity
ZeroEffHet=c()
for(i in c(0:10)){
logRORtrue=rnorm(1000,log(1),0.001+i*0.04)
logOR1=rnorm(1000,log(1),sqrt(0.0625))
logOR2=logOR1-logRORtrue
#generate SEs
selogOR1=sample(dataS1$selogORU,1000,replace=T)
selogOR2=sample(dataS1$selogORF,1000,replace=T)
#pool subgroups
pooled=metagen(TE=c(logOR1,logOR2),seTE=c(selogOR1,selogOR2),byvar=c(1:1000,1
:1000),comb.random=F)$TE.fixed.w
#coining
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin

#no coining
logRoR=logOR1-logOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoR=metagen(logRoR,selogRoR)$TE.random

#the coined
logRoRcoin=invlogOR1-invlogOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledlogRoRcoin=metagen(logRoRcoin,selogRoR)$TE.random
ZeroEffHet=rbind.data.frame(ZeroEffHet,c(exp(pooledlogRoRcoin),exp(pooledlogR
oR), 0.001+i*0.04))
}
names(ZeroEffHet)=c("RoRcoin","RoRnocoin","heterlogROR")

with(ZeroEffHet,plot(heterlogROR,RoRcoin,lty=3, main="Figure 1: ROR using coi
ning as a function of the heterogeneity of the true logROR",sub="OR1=ROR=1",
xlab="Heterogeneity SD of logROR",ylab="RoR with coining"))
with(ZeroEffHet,lines(heterlogROR,RoRcoin))
abline(h=1,lty=3)
```

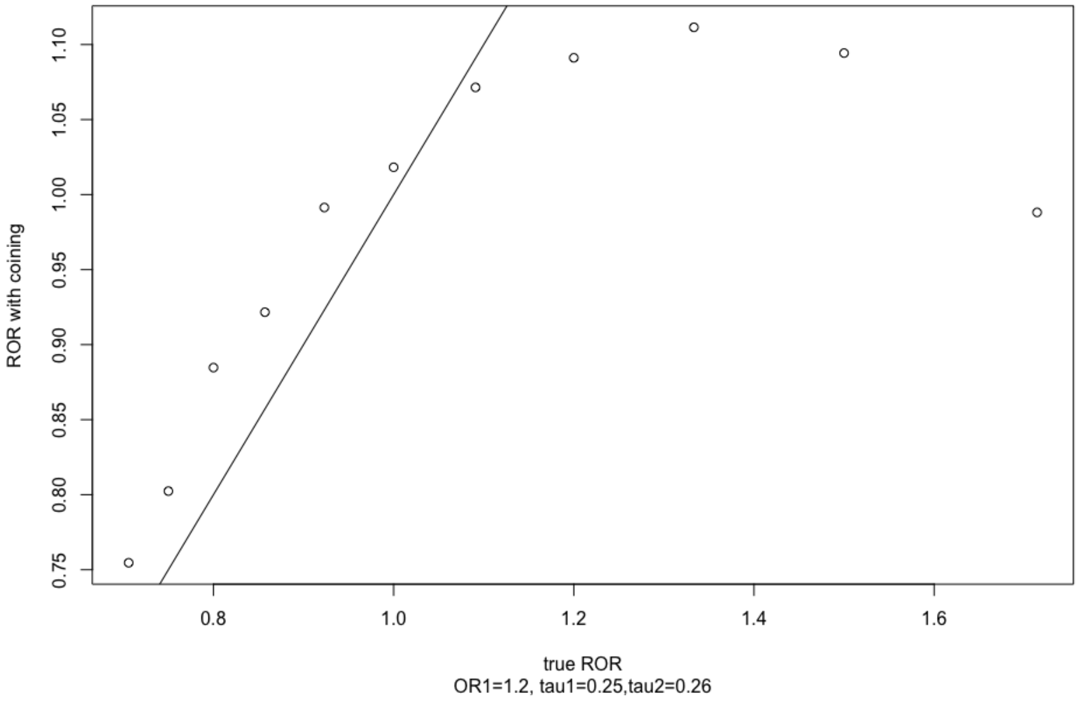
Figure 1: ROR using coining as a function of the heterogeneity of the true logROR



The followign graph summarizes the association between the underlying true ROR and the bias produced by coining. The presented line is the diagonal. We assumed that the two subgroup logOR have different heterogeneities (standard deviations 0.25 and 0.26 equal to those observed in the data by Evangelou et al.)

```
subdiffdiffobshet=c()
for(i in -5:5){

logOR1=rnorm(1000,log(1.2),0.25)
logOR2=rnorm(1000,log(1.2+i*0.1),0.26)
#generate SEs
selogOR1=sample(dataS1$selogORU,1000,replace=T)
selogOR2=sample(dataS1$selogORF,1000,replace=T)
#pool subgroups
pooled=metagen(TE=c(logOR1,logOR2),seTE=c(selogOR1,selogOR2),byvar=c(1:1000,1
:1000),comb.random=F)$TE.fixed.w
#coining
coin=(-as.numeric(pooled<0))+(as.numeric(pooled>0))
invlogOR1=logOR1*coin
invlogOR2=logOR2*coin

#without coining
logRoR=logOR1-logOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledRoR=metagen(logRoR,selogRoR)$TE.random
exp(pooledRoR)
#the coinedsqrt(0.06)
logRoRcoin=invlogOR1-invlogOR2
selogRoR=sqrt(selogOR1^2+selogOR2^2)
pooledRoRcoin=metagen(logRoRcoin,selogRoR)$TE.random
exp(pooledRoRcoin)

subdiffdiffobshet=rbind.data.frame(subdiffdiffobshet,c(exp(pooledRoRcoin),c(e
xp(pooledRoR)), 1.2/(1.2+i*0.1)))
}
names(subdiffdiffobshet)=c("RoRcoin", "RoRnocoin","ExpectedRoR")

with(subdiffdiffobshet,plot(ExpectedRoR,RoRcoin, main="Figure 2: ROR using co
ining as a function of the true ROR", sub="OR1=1.2, tau1=0.25,tau2=0.26", yla
b="ROR with coining",xlab="true ROR"))
abline(0,1)
```

Figure 2: ROR using coining as a function of the true ROR


ROR with coining
true ROR
OR1=1.2, tau1=0.25,tau2=0.26