Going beyond mean effect size; presenting prediction intervals for on-farm network trial analyses - Supplementary Material

Anabelle Laurent, Peter Kyveryga, Fernando Miguez, David Makowski

Required R packages and corresponding version

```
knitr::opts_chunk$set(echo = TRUE)
rm(list=ls())
library(dplyr)      # version 0.8.3
library(ggplot2)    # version 3.2.1
library(MCMCglmm)   # version 2.29
library(lme4)       # version 1.1-21
library(Hmisc)      # version 4.2-0
library(forcats)    # version 0.4.0
```

```
## You might want to adjust the path as needed
simdata<-read.csv("../data/simdata.csv")
```

For the purpose of the supplementary material, we used a data frame called "datasim" including simulated yield ratio (lrr) of a hypothetical management practice to a control for each replicate (rep) within one trial identifier (Trial_ID).

Please find the csv file at https://github.com/femiguez/EJA_OFRN_prediction_intervals.

```
head(simdata)
```

```
##   Trial_ID rep        lrr
## 1       T1   1 -1.5664706
## 2       T1   2 -0.2764872
## 3       T1   3 -0.7414822
## 4       T1   4  0.9210677
## 5       T1   5  0.8332006
## 6       T1   6  0.7745080
```

Frequentist approach using lme4 R package (Bates et al., 2015)

First, run the random-effect model

```
mod_fq=lmer(lrr ~ 1  + (1|Trial_ID),data= simdata)
```

- point estimate of log ratio (median)

```
fit = fixef(mod_fq)
```

- Lower bound of confidence interval (2.5%)

```
low = as.data.frame(confint(mod_fq))[3,1]
```

- Upper bound of confidence interval (97.5%)

```
up = as.data.frame(confint(mod_fq))[3,2]
```

- prediction interval based on the method of Higgins et al. (2009)

```
bs_var= (as.data.frame(VarCorr(mod_fq))[1,5])^2 # estimated between-trial variance
se = as.data.frame(coef(summary(mod_fq)))[1,2]  # standard error of the intercept
var = se^2                                      # variance of the estimated mean effect
k = length(unique(simdata$Trial_ID))            # number of trials
mu=fixef(mod_fq)                                # quantile 0.5 of the prediction interval
low = mu-qt(.975, df=k-2)*sqrt(bs_var+var)      # quantile 0.025 of the prediction interval
up = mu+qt(.975, df=k-2)*sqrt(bs_var+var)       # quantile 0.975 of the prediction interval
```

- probability of ineffective treatment

```
var_total<-sqrt(bs_var+var)  # total variability
q<-mu/var_total              # mean = q*total variability
pt(q, df=k-2)                # student t distribution
proba=1-pt(q, df=k-2)        # probability of ineffective treatment
```

Bayesian approach using MCMCglmm R package (Hadfield, 2010)

- credibility interval and individual trial yield response

First, run the random-effect model

```
prior1<- list(B = list(mu=0,V=2),              # prior on the fixed effect
              G=list(G1=list(V=1, nu=0.002)),  # prior on the between-trial variance
              R = list(V = 1, nu = 0.002))     # prior on the within-trial variance

mod_bayes<-MCMCglmm(lrr~1,      # fixed effect
             random=~Trial_ID,  # random effect
             data=simdata,
             family="gaussian",
             prior=prior1,      # priors (defined above)
             thin=10,           # thinning interval
             nitt= 200000,      # number of iterations
             burnin = 20000,    # burnin
             verbose=FALSE,     # if true MH diagnostics are printed to screen
             pr=TRUE,           # posterior distribution of random effects are saved
             DIC=TRUE)          # deviance information criterion
```

- credibility interval of the mean log ratio

```
cred_int<-predict(mod_bayes,interval="confidence",level=0.95)[1,]
```

fit represents the point estimate of log ratio (median)
lwr represents the lower bound of credibility interval (quantile 0.025)
upr represents the upper bound of credibility interval (quantile 0.975)

- credibility interval of individual log yield ratio

```
pred<-predict(mod_bayes,interval="confidence", marginal=mod_bayes$Trial_ID,level=0.95)
pred<-as.data.frame(pred)
pred$trial<-simdata$Trial_ID

fit<-pred$fit
lwr<-pred$lwr
```

```
upr<-pred$upr
pred$trial <- reorder(pred$trial, pred$fit)

duplication<-which(duplicated(pred))
pred<-pred[-duplication,]
print(pred)    # dataframe returning the credibility interval of individual log ratio
```

```
##             fit         lwr        upr trial
## 1    0.02722221 -0.62771592 0.69615182    T1
## 7    0.36658628 -0.31252249 1.02767121    T2
## 13  -0.02602462 -0.69257259 0.62242759    T3
## 19   0.92256398  0.25681759 1.65279184    T4
## 25   0.36579648 -0.28833387 1.03680107    T5
## 31  -0.64354286 -1.33255553 0.04474673    T6
## 37   0.27715138 -0.39855676 0.93898269    T7
## 43  -0.01641640 -0.70555867 0.63115466    T8
## 49  -0.17090482 -0.83345405 0.48333606    T9
## 55  -0.27320065 -0.95197036 0.38540348   T10
## 61  -0.52190826 -1.23108737 0.14032813   T11
## 67  -0.47672252 -1.16094063 0.21415137   T12
## 73   1.05423963  0.34543264 1.75527089   T13
## 79  -0.65407695 -1.37499138 0.03630317   T14
## 85   0.54730588 -0.10621101 1.23771108   T15
## 91   0.68788347  0.01316752 1.37195944   T16
## 97   0.19070020 -0.47133804 0.86444651   T17
## 103 -0.19840422 -0.87063133 0.46147629   T18
## 109  0.88321530  0.17839057 1.57721851   T19
## 115 -0.68197285 -1.37004737 0.01227102   T20
## 121  0.94894026  0.24088809 1.62438932   T21
## 127 -0.16100843 -0.82150817 0.51735748   T22
## 133  0.23039292 -0.44952318 0.87695845   T23
## 139  0.66442806 -0.02377980 1.33111082   T24
## 145 -0.40628059 -1.08893619 0.26586178   T25
```

The column "fit" represents the point estimate of log ratio for each individual trial (see column "trial") (median)
The column "lwr" represents the lower bound of credibility interval for each individual trial (quantile 0.025)
The column "upr" represents the upper bound of credibility interval for each individual trial (quantile 0.975)

- prediction interval for a new trial

In order to compute a prediction interval, you need to add a new trial with missing value was added (here called IDNew) and run the random effect model.

```
simdata<-simdata %>% add_row(Trial_ID="IDNew")
tail(simdata)
```

```
##     Trial_ID rep         lrr
## 146      T25   2 -0.75740552
## 147      T25   3 -0.01329639
## 148      T25   4 -0.61918290
## 149      T25   5  0.38288358
## 150      T25   6 -0.59204070
## 151    IDNew  NA          NA
```

```
mod_bayes_new<-MCMCglmm(lrr~1,  # fixed effect
             random=~Trial_ID,  # random effect
             data=simdata,
             family="gaussian",
             prior=prior1,      # priors (defined above)
             thin=10,           # thinning interval
             nitt= 200000,      # number of iterations
             burnin = 20000,    # burnin
             verbose=FALSE,     # if true MH diagnostics are printed to screen
             pr=TRUE,           # posterior distribution of random effects are saved
             DIC=TRUE)
```

Instead of using the function predict, the MCMC chains were used to compute the prediction interval

```
REnew<-as.data.frame(mod_bayes_new$Sol[,"Trial_ID.IDNew"]) # MCMC chain for the random effect IDNew
intercept<-as.data.frame(mod_bayes_new$Sol[,"(Intercept)"]) # MCMC chain for the the intercept
newtrial<-REnew+intercept                                  # sum the two MCMC chains
predinterval<-quantile(newtrial[,1], c(.025, .5, .975))  # get quantiles
```

- probability of ineffective treatment

```
X<-sum(newtrial$var1<0)   # number of cases where log ratio <0
n<-length(newtrial$var1)  # total number of possible cases
proba<-X/n                # probability of ineffective treatment
```

You do need an exponential transformation to convert the Frequentist and Bayesian outputs from log yield ratio to yield ratio.

Figures

We use the exponential transformation for the figures

```
# outputs = dataframe combining trial estimations, credible interval and prediction interval
outputs <- pred %>%
  add_row(fit=cred_int[1],
          lwr=cred_int[2],
          upr=cred_int[3],
          trial="mean yield ratio") %>%
  add_row(fit=predinterval[2],
          lwr=predinterval[1],
          upr=predinterval[3],
          trial="prediction interval")

outputs<- outputs %>%
  mutate(trial = forcats::fct_reorder(trial,fit)) %>%
  mutate(trial = forcats::fct_relevel(trial, "prediction interval", after = 0)) %>%
  mutate(trial = forcats::fct_relevel(trial, "mean yield ratio", after = 1))
```

```
ggplot(outputs, aes(x = exp(fit), xmin = exp(lwr), xmax = exp(upr), y = trial )) +
  geom_point(size=2) +
  geom_segment( aes(x = exp(lwr), xend = exp(upr), y = trial, yend=trial)) +
  theme_bw() +
  xlab("Yield ratio") +
  ylab("") +
  geom_text(size=5,aes(x=5.2,y=1.2,label= ifelse(trial=="prediction interval",
                                    round(proba,2),""))) +
```

```
  geom_vline(xintercept = 1, color = "black", size=0.8) +
  theme(axis.text=element_text(size=12,face="bold"),
        axis.title =element_text(size=12,face="bold"),
        legend.text=element_text(size=12,face="bold"),
        legend.title=element_blank())
```

References

Bates, D., Mächler, M., Bolker, B., Walker, S., 2015. Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software 67. https://doi.org/10.18637/jss.v067.i01

Hadfield, J.D., 2010. MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package. Journal of Statistical Software 33. https://doi.org/10.18637/jss.v033.i02

Higgins, J.P.T., Thompson, S.G., Spiegelhalter, D.J., 2009. A re-evaluation of random-effects meta-analysis. Journal of the Royal Statistical Society: Series A (Statistics in Society) 172, 137–159. https://doi.org/10.1111/j.1467-985X.2008.00552.x

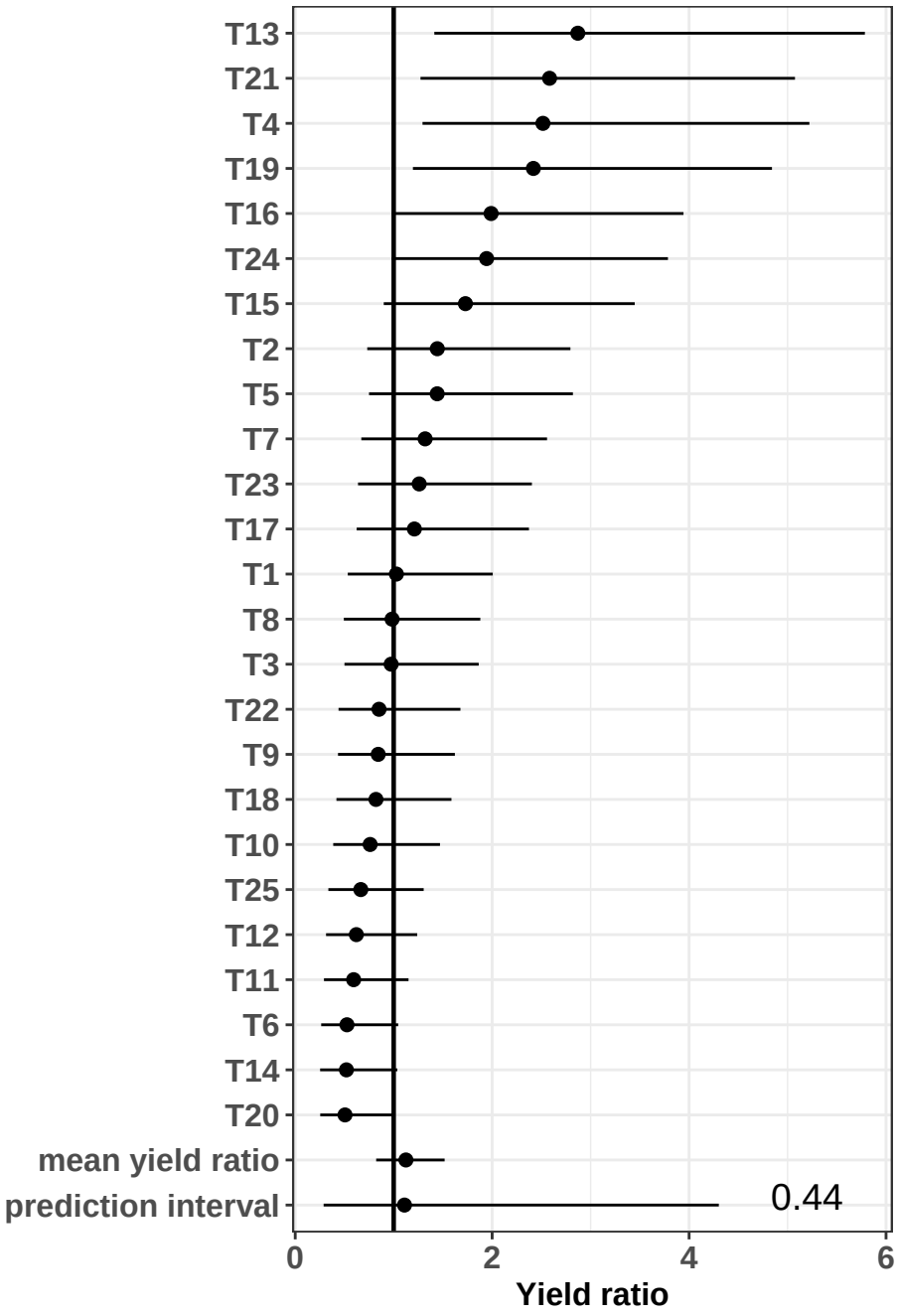

Figure 1: Estimated individual effects (denoted by T and a numerical value) with their 95% credible intervals, mean yield ratio and its 95% credible interval, and 95% prediction interval. The numerical value at the bottom-right indicates the probability of ineffective treatment (probability of yield ratio less than 1) in a new trial