Theory and Practice of random effects and REML in *variancePartition* and *dream*

Gabriel Hoffman

Pamela Sklar Division of Psychiatric Genomics
Icahn Institute for Data Science and Genomic Technology
Department of Genetics and Genomic Sciences
Icahn School of Medicine at Mount Sinai

August 7, 2020

The distinction between modeling a variable as a fixed versus a random effect depends on the goal of the statistical analysis. While some theory and software make a strong distinction, *variancePartition* and *dream* take different approaches based on the goal of each type of analysis.

Here we consider the distinction between fixed and random effects, and the usage of REML in *variancePartition* and *dream*.

1 *variancePartition*: estimating contributions to expression variation

In traditional statistics and biostatistics, there is a strong distinction between modeling categorical variants as fixed and random effects. Random effects correspond to a sample of units from a larger population, while fixed effects correspond to properties of specific individuals. Random effects are typically treated as nuisance variables and integrated out, and hypothesis testing is performed on the fixed effect.

The *r2glmm* package fits into this traditional framework, by computing the variance fractions for a given fixed effect as:

$$\sigma^2_{fixed}/(\sigma^2_{fixed} + \sigma^2_{error}) \quad \boxed{1}$$

Importantly, the random effects are not in the denominator. The fraction is only determined by fixed effects and residuals.

In my experience in bioinformatics, this was a problem. Making such distinctions between fixed and random effects seemed arbitrary. Variance in a phenotype could be due to age (fixed) or to variation across subject (random). Including all of the variables in the denominator produced more intuitive results so that 1) the variance fractions sum to one across all components and 2) fixed and random effects could be interpreted on the same scale 3) fractions could be compared across studies with different designs, 4) estimates of variance fractions were most accurate. So in variancePartition the fractions are defined as:

$$\sigma^2_X/(\sigma^2_{fixed} + \sigma^2_{random} + \sigma^2_{error}) \quad \boxed{2}$$

just plugging the each variable in the numerator.

Thus the faction evaluated by variancePartition is different than *r2glmm* by definition.

Here is some code explicitly demonstrating this difference:

```
library('variancePartition')
library('lme4')
library('r2glmm')

set.seed(1)

N = 1000
beta = 3
```

```
alpha = c(1, 5, 7)

# generate 1 fixed variable and 1 random variable with 3 levels
data = data.frame(X=rnorm(N), Subject = sample(c('A', 'B', 'C'), 100, replace=TRUE))

# simulate variable
# y = X\beta + Subject\alpha + \sigma^2
data$y = data$X*beta + model.matrix(~ data$Subject) %*% alpha  + rnorm(N, 0, 1)

# fit model
fit = lmer( y ~ X +(1|Subject), data, REML=FALSE)

# calculate variance fraction using variancePartition
# include the total sum in the denominator
frac = calcVarPart(fit)
frac
```

```
  Subject         X Residuals
   0.4505    0.4952    0.0543
```

```
# the variance fraction excluding the random effect from the denominator
# is the same as from r2glmm
frac[['X']] / (frac[['X']] + frac[['Residuals']])
```

```
[1] 0.901
```

```
# using r2glmm
r2beta(fit)
```

```
  Effect   Rsq upper.CL lower.CL
1  Model 0.896    0.904    0.886
2      X 0.896    0.904    0.886
```

So the formulas are different. But why require categorical variables as random effects?

At practical level, categorical variables with too many levels are problematic. Using a categorical variable with 200 categories as a fixed effect is statistically unstable. There are so many degrees of freedom that that variable will absorb a lot of variance even under the null. Statistically, estimating the variance fraction for a variable with many categories can be biased if that variable is a fixed effect. Therefore, *variancePartition* requires all categorical variables to be random effects. Modeling this variable as a random effect produces unbiased estimates of variance fractions in practice. See simulations in the Supplement (section 1.5) of Hoffman and Schadt (2016).

The distinction between fixed and random effects is important in the *r2glmm* formulation because it affects which variables are put in the denominator. So choosing to model a variable as a fixed versus random effect will definitely change the estimated fraction.

Yet for the *variancePartition* formulation, all variables are in the denominator and it isn't affected by the fixed/random decision. Moreover, using a random effect empirically reduces the bias of the estimated fraction.

Finally, why use maximum likelihood to estimate the paramters instead of the default REML (`REML=FALSE`)? Maximum likelihood fits all parameters jointly so that it estimates the fixed and random effects together. This is essential if we want to compare fixed and random effects later. Conversely, REML estimates the random effects by removing the fixed effects from the response before estimation. This implicitly removes the fixed effects from the denominator when evaluating the variance fraction. REML treats fixed effects as nuisance variables, while *variancePartition* considers fixed effects to be a core part of the analysis.

While REML produced unbiased estimates of the variance components, the goal of *variancePartition* is to estimate the variance fractions for fixed and random effects jointly. In simulations from the Supplement (section 1.5) of Hoffman and Schadt (2016), REML produced biased estimates of the variance fractions while maximum likelihood estimates are unbiased.

2 *dream*: hypothesis testing

While *dream* is also based on a linear mixed model, the goal of this analysis is to perform hypothesis testing on fixed effects. Random effects are treated as nuisance variables to be integrated out, and the approximate null distribution of a t- or F-statistic is constructed from the model fit.

Since the goal of the analysis is different, the consideration of using REML versus ML is different than above. While `REML=TRUE` is required by `lmerTest` called by `dream` when `ddf='Kenward-Roger'`, `ddf='Satterthwaite'` can be used with `REML` as either `TRUE` or `FALSE`. Since the Kenward-Roger method gave the best power with an accurate control of false positive rate in our simulations, and since the Satterthwaite method with `REML=TRUE` gives p-values that are slightly closer to the Kenward-Roger p-values, `REML=TRUE` is set as the default.