

Translating lme4 models to sommer

Giovanny Covarrubias-Pazaran

2021-05-05

The sommer package was developed to provide R users a powerful and reliable multivariate mixed model solver. The package is focused on problems of the type $p > n$ (more effects to estimate than observations) and its core algorithm is coded in C++ using the Armadillo library. This package allows the user to fit mixed models with the advantage of specifying the variance-covariance structure for the random effects, and specifying heterogeneous variances, and obtaining other parameters such as BLUPs, BLUEs, residuals, fitted values, variances for fixed and random effects, etc.

The purpose of this vignette is to show how to translate the syntax formula from lme4 models to sommer models. Feel free to remove the comment marks from the lme4 code so you can compare the results.

- 1) Random slopes with same intercept
- 2) Random slopes and random intercepts (without correlation)
- 3) Random slopes and random intercepts (with correlation)
- 4) Random slopes with a different intercept
- 5) Other models not available in lme4

1) Random slopes

This is the simplest model people use when a random effect is desired and the levels of the random effect are considered to have the same intercept.

```
# install.packages("lme4")
# library(lme4)
library(sommer)
data(DT_sleepstudy)
DT <- DT_sleepstudy
#####
## lme4
#####
# fm1 <- lmer(Reaction ~ Days + (1 | Subject), data=DT)
# summary(fm1) # or vc <- VarCorr(fm1); print(vc, comp=c("Variance"))
# Random effects:
# Groups   Name          Variance Std.Dev.
# Subject (Intercept) 1378.2   37.12
# Residual                960.5   30.99
# Number of obs: 180, groups: Subject, 18
#####
## sommer
#####
fm2 <- mmer(Reaction ~ Days,
            random= ~ Subject,
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp
```

```
##                               VarComp VarCompSE  Zratio Constraint
## Subject.Reaction-Reaction 1377.9758  505.0776  2.728246   Positive
## units.Reaction-Reaction   960.4705  107.0638  8.971013   Positive
```

2) Random slopes and random intercepts (without correlation)

This is the a model where you assume that the random effect has different intercepts based on the levels of another variable. In addition the || in lme4 assumes that slopes and intercepts have no correlation.

```
#####
## lme4
#####
# fm1 <- lmer(Reaction ~ Days + (Days || Subject), data=DT)
# summary(fm1) # or vc <- VarCorr(fm1); print(vc, comp=c("Variance"))
# Random effects:
# Groups   Name          Variance Std.Dev.
# Subject  (Intercept) 627.57  25.051
# Subject.1 Days         35.86   5.988
# Residual                   653.58  25.565
# Number of obs: 180, groups: Subject, 18
#####
## sommer
#####
fm2 <- mmer(Reaction ~ Days,
            random= ~ Subject + vs(Days, Subject),
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp
```

```
##                               VarComp VarCompSE  Zratio Constraint
## Subject.Reaction-Reaction   627.54087 283.52939 2.213319   Positive
## Days:Subject.Reaction-Reaction 35.86008 14.53187 2.467686   Positive
## units.Reaction-Reaction     653.58305 76.72711 8.518281   Positive
```

Notice that Days is a numerical (not factor) variable.

3) Random slopes and random intercepts (with correlation)

This is the a model where you assume that the random effect has different intercepts based on the levels of another variable. In addition a single | in lme4 assumes that slopes and intercepts have a correlation to be estimated.

```
#####
## lme4
#####
# fm1 <- lmer(Reaction ~ Days + (Days | Subject), data=DT)
# summary(fm1) # or # vc <- VarCorr(fm1); print(vc, comp=c("Variance"))
# Random effects:
# Groups   Name          Variance Std.Dev. Corr
# Subject  (Intercept) 612.10  24.741
#          Days         35.07   5.922  0.07
# Residual                   654.94  25.592
# Number of obs: 180, groups: Subject, 18
#####
## sommer
```

```
#####
## no equivalence in sommer to find the correlation between the 2 vc
## this is the most similar which is equivalent to (intercept || slope)
fm2 <- mmer(Reaction ~ Days,
            random= ~ Subject + vs(Days, Subject),
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp

##
## VarComp VarCompSE Zratio Constraint
## Subject.Reaction-Reaction 627.54087 283.52939 2.213319 Positive
## Days:Subject.Reaction-Reaction 35.86008 14.53187 2.467686 Positive
## units.Reaction-Reaction 653.58305 76.72711 8.518281 Positive
```

4) Random slopes with a different intercept

This is the a model where you assume that the random effect has different intercepts based on the levels of another variable but there's not a main effect. The 0 in the intercept in lme4 assumes that random slopes interact with an intercept but without a main effect.

```
#####
## lme4
#####
# fm1 <- lmer(Reaction ~ Days + (0 + Days | Subject), data=DT)
# summary(fm1) # or vc <- VarCorr(fm1); print(vc, comp=c("Variance"))
# Random effects:
# Groups Name Variance Std.Dev.
# Subject Days 52.71 7.26
# Residual 842.03 29.02
# Number of obs: 180, groups: Subject, 18
#####
## sommer
#####
fm2 <- mmer(Reaction ~ Days,
            random= ~ vs(Days, Subject),
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp

##
## VarComp VarCompSE Zratio Constraint
## Days:Subject.Reaction-Reaction 52.70946 19.09984 2.759681 Positive
## units.Reaction-Reaction 842.02736 93.84640 8.972399 Positive
```

4) Other models available in sommer but not in lme4

One of the strengths of sommer is the availability of other variance covariance structures. In this section we show 4 models available in sommer that are not available in lme4 and might be useful.

```
library(orthopolynom)
## diagonal model
fm2 <- mmer(Reaction ~ Days,
            random= ~ vs(ds(Daysf), Subject),
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp

##
## VarComp VarCompSE Zratio Constraint
```

```
## 0:Subject.Reaction-Reaction 139.5473 399.5095 0.3492967 Positive
## 1:Subject.Reaction-Reaction 196.8544 411.8262 0.4780037 Positive
## 2:Subject.Reaction-Reaction 0.0000 365.3178 0.0000000 Positive
## 3:Subject.Reaction-Reaction 556.0773 501.2665 1.1093445 Positive
## 4:Subject.Reaction-Reaction 855.2104 581.8190 1.4698910 Positive
## 5:Subject.Reaction-Reaction 1699.4269 820.4561 2.0713197 Positive
## 6:Subject.Reaction-Reaction 2910.8975 1175.7872 2.4757011 Positive
## 7:Subject.Reaction-Reaction 1539.6201 779.1437 1.9760413 Positive
## 8:Subject.Reaction-Reaction 2597.5337 1089.4522 2.3842568 Positive
## 9:Subject.Reaction-Reaction 3472.7108 1351.5702 2.5693899 Positive
## units.Reaction-Reaction 879.6958 247.4680 3.5547862 Positive
```

```
## unstructured model
fm2 <- mmer(Reaction ~ Days,
            random= ~ vs(us(Daysf), Subject),
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp
```

##		VarComp	VarCompSE	Zratio	Constraint
## 0:	Subject.Reaction-Reaction	402.6286	572.0867	0.7037894	Positive
## 1:0:	Subject.Reaction-Reaction	1022.5098	393.6922	2.5972314	Unconstr
## 1:	Subject.Reaction-Reaction	417.6460	521.3722	0.8010515	Positive
## 2:0:	Subject.Reaction-Reaction	540.3746	287.1704	1.8817210	Unconstr
## 2:1:	Subject.Reaction-Reaction	828.5156	325.7576	2.5433499	Unconstr
## 2:	Subject.Reaction-Reaction	0.0000	509.8962	0.0000000	Positive
## 3:0:	Subject.Reaction-Reaction	798.3750	397.0884	2.0105726	Unconstr
## 3:1:	Subject.Reaction-Reaction	1137.3863	443.9056	2.5622256	Unconstr
## 3:2:	Subject.Reaction-Reaction	1057.0708	385.9026	2.7392162	Unconstr
## 3:	Subject.Reaction-Reaction	760.2469	436.7463	1.7407060	Positive
## 4:0:	Subject.Reaction-Reaction	757.8909	411.2464	1.8429119	Unconstr
## 4:1:	Subject.Reaction-Reaction	1039.6832	447.5192	2.3232148	Unconstr
## 4:2:	Subject.Reaction-Reaction	911.1369	377.9651	2.4106377	Unconstr
## 4:3:	Subject.Reaction-Reaction	1590.6778	566.5376	2.8077180	Unconstr
## 4:	Subject.Reaction-Reaction	957.1797	364.0599	2.6291817	Positive
## 5:0:	Subject.Reaction-Reaction	932.5247	516.7169	1.8047110	Unconstr
## 5:1:	Subject.Reaction-Reaction	1179.5219	547.9498	2.1526095	Unconstr
## 5:2:	Subject.Reaction-Reaction	859.1635	440.5250	1.9503173	Unconstr
## 5:3:	Subject.Reaction-Reaction	1672.9989	664.0846	2.5192556	Unconstr
## 5:4:	Subject.Reaction-Reaction	2003.0167	738.6399	2.7117633	Unconstr
## 5:	Subject.Reaction-Reaction	2067.9299	553.3254	3.7372765	Positive
## 6:0:	Subject.Reaction-Reaction	666.1077	565.7589	1.1773702	Unconstr
## 6:1:	Subject.Reaction-Reaction	850.9395	583.6190	1.4580394	Unconstr
## 6:2:	Subject.Reaction-Reaction	916.2375	504.0273	1.8178333	Unconstr
## 6:3:	Subject.Reaction-Reaction	1785.8432	750.7274	2.3788171	Unconstr
## 6:4:	Subject.Reaction-Reaction	2077.5064	822.0777	2.5271412	Unconstr
## 6:5:	Subject.Reaction-Reaction	2603.2823	1035.1406	2.5149070	Unconstr
## 6:	Subject.Reaction-Reaction	3123.2005	1049.0352	2.9772123	Positive
## 7:0:	Subject.Reaction-Reaction	932.8190	490.4744	1.9018709	Unconstr
## 7:1:	Subject.Reaction-Reaction	927.3416	492.7764	1.8818709	Unconstr
## 7:2:	Subject.Reaction-Reaction	924.7079	426.2387	2.1694602	Unconstr
## 7:3:	Subject.Reaction-Reaction	1282.8637	583.3415	2.1991642	Unconstr
## 7:4:	Subject.Reaction-Reaction	1549.9053	643.7083	2.4077757	Unconstr
## 7:5:	Subject.Reaction-Reaction	1941.5523	811.3286	2.3930529	Unconstr
## 7:6:	Subject.Reaction-Reaction	2306.0261	951.5128	2.4235367	Unconstr
## 7:	Subject.Reaction-Reaction	1669.8274	612.0081	2.7284398	Positive

```
## 8:0:Subject.Reaction-Reaction  920.3110  576.8500  1.5954079  Unconstr
## 8:1:Subject.Reaction-Reaction  1044.9313  592.5243  1.7635247  Unconstr
## 8:2:Subject.Reaction-Reaction   831.4993  486.9625  1.7075221  Unconstr
## 8:3:Subject.Reaction-Reaction  1607.0156  717.6871  2.2391591  Unconstr
## 8:4:Subject.Reaction-Reaction  2029.1022  805.6724  2.5185201  Unconstr
## 8:5:Subject.Reaction-Reaction  3058.1945 1093.4722  2.7967739  Unconstr
## 8:6:Subject.Reaction-Reaction  2927.6051 1177.5589  2.4861644  Unconstr
## 8:7:Subject.Reaction-Reaction  2433.2427  957.7103  2.5406876  Unconstr
## 8:Subject.Reaction-Reaction    2947.1635  844.8113  3.4885466  Positive
## 9:0:Subject.Reaction-Reaction  1440.6886  690.1726  2.0874323  Unconstr
## 9:1:Subject.Reaction-Reaction  1514.9679  703.4423  2.1536491  Unconstr
## 9:2:Subject.Reaction-Reaction   967.8504  550.1628  1.7592073  Unconstr
## 9:3:Subject.Reaction-Reaction  1742.6866  797.5934  2.1849310  Unconstr
## 9:4:Subject.Reaction-Reaction  2198.3504  892.7701  2.4623924  Unconstr
## 9:5:Subject.Reaction-Reaction  3236.8715 1196.2341  2.7058847  Unconstr
## 9:6:Subject.Reaction-Reaction  2210.6321 1185.1233  1.8653182  Unconstr
## 9:7:Subject.Reaction-Reaction  2399.5130 1027.8125  2.3345824  Unconstr
## 9:8:Subject.Reaction-Reaction  3847.0132 1391.5584  2.7645359  Unconstr
## 9:Subject.Reaction-Reaction    3946.2369 1228.6678  3.2118013  Positive
## units.Reaction-Reaction        883.2477  577.9203  1.5283210  Positive
```

```
## random regression (legendre polynomials)
fm2 <- mmer(Reaction ~ Days,
            random= ~ vs(leg(Days,1), Subject),
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp
```

```
##                VarComp  VarCompSE  Zratio  Constraint
## leg0:Subject.Reaction-Reaction 2817.4048 1011.23903 2.786092  Positive
## leg1:Subject.Reaction-Reaction  473.4608  199.53635 2.372805  Positive
## units.Reaction-Reaction        654.9433   77.18822 8.485016  Positive
```

```
## unstructured random regression (legendre)
fm2 <- mmer(Reaction ~ Days,
            random= ~ vs(us(leg(Days,1)), Subject),
            data=DT, tolparinv = 1e-6, verbose = FALSE)
summary(fm2)$varcomp
```

```
##                VarComp  VarCompSE  Zratio  Constraint
## leg0:Subject.Reaction-Reaction 2817.4056 1011.24156 2.786086  Positive
## leg1:leg0:Subject.Reaction-Reaction 869.9590  381.02481 2.283208  Unconstr
## leg1:Subject.Reaction-Reaction   473.4608  199.53612 2.372807  Positive
## units.Reaction-Reaction        654.9428   77.18763 8.485075  Positive
```

Final remarks

Keep in mind that `sommer` uses the direct inversion (DI) algorithm which can be very slow for large datasets. The package is focused on problems of the type $p > n$ (more random effect levels than observations) and models with dense covariance structures. For example, for an experiment with dense covariance structures with low-replication (i.e. 2000 records from 1000 individuals replicated twice with a covariance structure of 1000x1000), `sommer` will be faster than MME-based software. Also for genomic problems with large number of random effect levels, i.e. 300 individuals (n) with 100,000 genetic markers (p). For highly replicated trials with small number of individuals and covariance structures or $n > p$ (i.e. 2000 records from 200 individuals replicated 10 times with covariance structure of 200x200), `asrem1` or other MME-based algorithms will be much faster and we recommend you to opt for those software programs.

Literature

- Covarrubias-Pazarán G. 2016. Genome assisted prediction of quantitative traits using the R package sommer. PLoS ONE 11(6):1-15.
- Covarrubias-Pazarán G. 2018. Software update: Moving the R package sommer to multivariate mixed models for genome-assisted prediction. doi: <https://doi.org/10.1101/354639>
- Bernardo Rex. 2010. Breeding for quantitative traits in plants. Second edition. Stemma Press. 390 pp.
- Gilmour et al. 1995. Average Information REML: An efficient algorithm for variance parameter estimation in linear mixed models. Biometrics 51(4):1440-1450.
- Henderson C.R. 1975. Best Linear Unbiased Estimation and Prediction under a Selection Model. Biometrics vol. 31(2):423-447.
- Kang et al. 2008. Efficient control of population structure in model organism association mapping. Genetics 178:1709-1723.
- Lee, D.-J., Durban, M., and Eilers, P.H.C. (2013). Efficient two-dimensional smoothing with P-spline ANOVA mixed models and nested bases. Computational Statistics and Data Analysis, 61, 22 - 37.
- Lee et al. 2015. MTG2: An efficient algorithm for multivariate linear mixed model analysis based on genomic information. Cold Spring Harbor. doi: <http://dx.doi.org/10.1101/027201>.
- Maier et al. 2015. Joint analysis of psychiatric disorders increases accuracy of risk prediction for schizophrenia, bipolar disorder, and major depressive disorder. Am J Hum Genet; 96(2):283-294.
- Rodríguez-Alvarez, María Xose, et al. Correcting for spatial heterogeneity in plant breeding experiments with P-splines. Spatial Statistics 23 (2018): 52-71.
- Searle. 1993. Applying the EM algorithm to calculating ML and REML estimates of variance components. Paper invited for the 1993 American Statistical Association Meeting, San Francisco.
- Yu et al. 2006. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Genetics 38:203-208.
- Tunncliffe W. 1989. On the use of marginal likelihood in time series model estimation. JRSS 51(1):15-27.