

This material has been published in CAB Reviews, issue 7, No. 016, 2012, doi: 10.1079/PAVSNNR20127020, the only accredited archive of the content that has been certified and accepted after peer review. Copyright and all rights therein are retained by CABI.

The electronic version of this article is the definitive one. It is located here: <http://www.cabi.org/cabreviews>

Potential of GLMM in modelling invasive spread: Supplement D

Analysing time series data of phenotypic measurements – trade-off in sexual reproduction and vegetative growth of perennial ryegrass (*Lolium perenne*)?

J. Thiele^{1*} and B. Markussen²

Address: ¹ Institute of Landscape Ecology, University of Muenster, Robert-Koch-Str. 28, 48149 Muenster, Germany. ² Department of Basic Sciences and Environment, University of Copenhagen, Thorvaldsensvej 40, 1871 Frederiksberg C, Denmark.

*Correspondence: J. Thiele. Email: jan.thiele@uni-muenster.de

Introduction

In this example, we analyse repeated phenotypic measurements of perennial ryegrass (*Lolium perenne*) from a greenhouse experiment. We hypothesize that there is a trade-off between sexual reproduction (flower stems) and vegetative propagation (tillers) within polycormons of this species. This is not directly a study of biological invasions, however, we want to find out, if less flowering plants (cultivars, in this case) would be more competitive and could invade seminatural grasslands. Similar repeated measures designs are quite often used in invasion biology. We acknowledge that we adopted some ideas from an example presented by Fox (2002) for this analysis.

Sexual reproduction or, more precisely, the degree of flowering of the plants, was modified by cold treatments (vernalisation). Cold temperature (< 6° C) are a prerequisite for flowering of perennial ryegrass. The shorter the cold period, the lower the number of flowering stems that emerge afterwards. In this way, plants (polycormons) of perennial ryegrass with varying numbers of flowering stems were produced.

The plants were grown from seeds that were collected from six populations and, when they had produced several tillers, they were split into four ‘clones’ of three tillers in order to replicate each individual genotype in four cold treatments. Two clones from different populations were planted together in a pot in order to create a competitive situation among plants. The same genotype combinations were repeated in the four cold treatments. The experiment comprised 640 pots, because there were 8 combinations of populations of the competing clones, and the population combinations were repeated 20 times (with different genotype combinations) in each of 4 cold treatments. As the pots contained two clones which were investigated at three time points, the whole dataset has got a sample size of 3840, but the effective sample size is 3777 because of some missing data.

At each of the three time points, the diameter of clones (i.e. polycormon diameter) was measured and the number of flowering stems was counted. The latter will be used as a predictor variable, because it was manipulated, and the clone diameter will be used as the response (dependent variable). The diameter of the other plant in the pot will be used as a covariate that represents the amount of competition. More details on the experimental setup can be found in Thiele et al. (2009).

For this analysis we will use the packages *nlme* (version 3.1-98; Pinheiro et al. 2011) and *lattice* (0.19-17; Sarkar 2008) in R (R Development Core Team 2011).

The dataset

We take a look at the structure of the dataset

```
> str(Lolium)

'data.frame':  3777 obs. of  9 variables:
 $ plantid : Factor w/ 1280 levels "p1","p10","p100",...: 1 393 504 615
726.
 $ pot      : Factor w/ 640 levels "pot1","pot10",...: 1 112 223 334 445
556
 $ gentyp  : Factor w/ 315 levels "Bacal","Bacal0",...: 1 34 12 17 21
32 30
 $ pop     : Factor w/ 6 levels "Baca","Bile",...: 1 1 1 1 1 1 1 1 1 1
...
 $ time    : Factor w/ 3 levels "t1","t2","t3": 1 1 1 1 1 1 1 1 1 1
...
 $ vern    : Factor w/ 4 levels "a","b","c","d": 1 1 1 1 1 1 1 1 1 1
...
 $ flowers : num  0 0 0 0 0 0 0 0 0 0 ...
 $ diam    : num  4 3 4 4 4 3 4 5 4 5.5 ...
 $ codiam  : num  4 4.5 4 4 3 3.5 3.5 3.5 2.5 3 ...
```

and see that there are 1280 plant individuals (`plantid`) grown in 640 pots (`pot`). The plants originate from 315 genotypes (`gentyp`) from 6 different populations (`pop`). We are most interested in the relationship between the diameter of clones (`diam`) and the number of flowers that they produced (`flowers`). We also need to take into account that the plants were measured at three time points. Here, we code `time` as a categorical variable with three levels. Finally, there is the diameter of the other clone in the pot (`codiam`), the competition co-variate.

The plants were kept at cold temperatures for different length of time to induce different degrees of flowering. This vernalisation treatment (`vern`) is likely to have affected the vegetative growth, too. We take a look at this using box-and-whisker plots

```
> library(lattice)

> bwplot(diam ~ vern|time, data= Lolium, layout=c(3,1),
xlab="Vernalisation treatment", ylab="Clone diameter" )
```

and see that clones in the longer vernalisation treatments (c, d) indeed have somewhat lower medians of diameter (Fig. 1). One way to tackle this would be to include `vern` into the model as another co-factor. However, as we are not interested in the vernalisation treatment itself, but only in the varying degree of flowering that it produced, we decide to center clone diameter (i.e. subtract the mean) within vernalisation treatment and year and to use these centered values `cdiam` and `ccodiam` in further analysis.

Pre-analyses

Before starting with Linear Mixed Modelling (LMM), we want to inspect the growth of the clones over the three time points graphically and see, if there are differences between flowering and non-flowering clones and between populations. As there are too many plants (1280) to display all of them in one graph, we draw a random sample of

plants stratified according to population and flowering status at the first time point. For this purpose, we create a subset of the `Lolium` data table that includes only data from the first time point

```
> Lolium.t1<- subset(Lolium, time=="t1" )
```

then we classify the plants into three classes of flowering (none, medium, many flowers)

```
> Lolium.t1$flclass<- cut(Lolium.t1$flowers, c(-1, 0, 20, 200), labels = c("none", "medium", "many" ) )
```

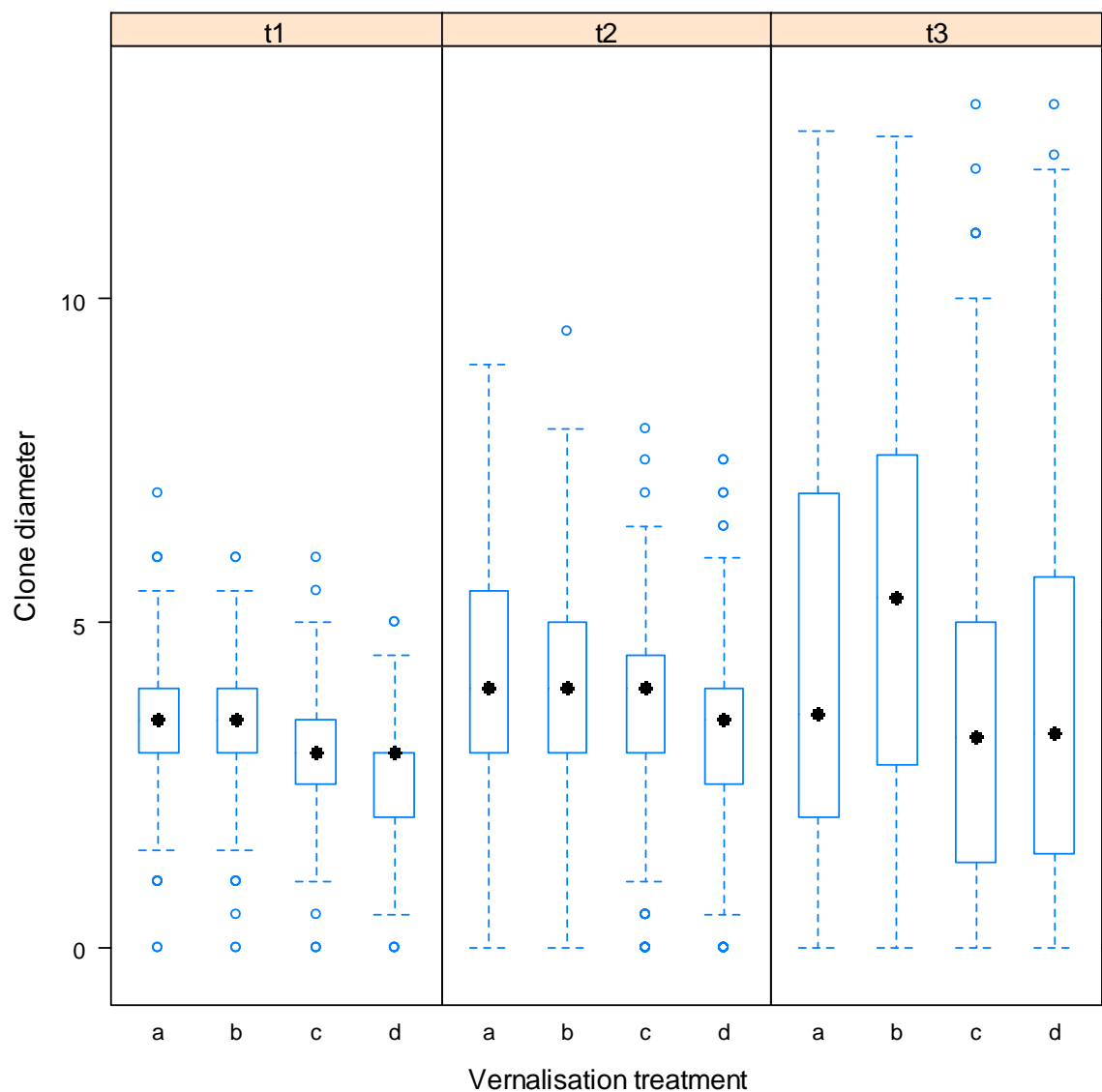


Figure 1 Diameter of clones (polycormons) vs. vernalisation treatment, i.e. different length of exposure to cold temperatures, for three time points of measurement (t1-3).

and create a grouping variable based on population and flowering class

```
> Lolium.t1$pop.flclass<- factor(Lolium.t1$pop:Lolium.t1$flclass)
```

that we will use for stratification.

Then, we randomly draw 10 plants from each combination of population and the flowering classes 'none' and 'many' (we omit medium flowering plants, to have a stronger contrast)

```
> attach(Lolium.t1)

> bolu.nofl.sample<- sample(unique(plantid[subset=
pop.flclass=="Bolu:none"]), 10)
> bolu.manyfl.sample<- sample(unique(plantid[subset=
pop.flclass=="Bolu:many"]), 10)
> sisu.nofl.sample<- sample(unique(plantid[subset=
pop.flclass=="Sisu:none"]), 10)
> sisu.manyfl.sample<- sample(unique(plantid[subset=
pop.flclass=="Sisu:many"]), 10)
> baca.nofl.sample<- sample(unique(plantid[subset=
pop.flclass=="Baca:none"]), 10)
> baca.manyfl.sample<- sample(unique(plantid[subset=
pop.flclass=="Baca:many"]), 10)
> bile.nofl.sample<- sample(unique(plantid[subset=
pop.flclass=="Bile:none"]), 10)
> bile.manyfl.sample<- sample(unique(plantid[subset=
pop.flclass=="Bile:many"]), 10)
> deru.nofl.sample<- sample(unique(plantid[subset=
pop.flclass=="Deru:none"]), 10)
> deru.manyfl.sample<- sample(unique(plantid[subset=
pop.flclass=="Deru:many"]), 10)
> veyo.nofl.sample<- sample(unique(plantid[subset=
pop.flclass=="Veyo:none"]), 10)
> veyo.manyfl.sample<- sample(unique(plantid[subset=
pop.flclass=="Veyo:many"]), 10)

> detach(Lolium.t1)
```

and combine the 12 strata (6 populations * 2 flowering classes) into one vector containing the plantid

```
> whole.sample<- append(as.vector(bolu.nofl.sample),
as.vector(bolu.manyfl.sample) )
whole.sample<- append(as.vector(whole.sample),
as.vector(sisu.nofl.sample) )
> whole.sample<- append(as.vector(whole.sample),
as.vector(sisu.manyfl.sample) )
whole.sample<- append(as.vector(whole.sample),
as.vector(baca.nofl.sample) )
> whole.sample<- append(as.vector(whole.sample),
as.vector(baca.manyfl.sample) )
whole.sample<- append(as.vector(whole.sample),
as.vector(bile.nofl.sample) )
> whole.sample<- append(as.vector(whole.sample),
as.vector(bile.manyfl.sample) )
whole.sample<- append(as.vector(whole.sample),
as.vector(deru.nofl.sample) )
> whole.sample<- append(as.vector(whole.sample),
as.vector(deru.manyfl.sample) )
whole.sample<- append(as.vector(whole.sample),
as.vector(veyo.nofl.sample) )
> whole.sample<- append(as.vector(whole.sample),
as.vector(veyo.manyfl.sample) )
```

which we use to create a subset of the data table `Lolium` that contains only the plants in the sample

```
> Lolium.sample<- Lolium[is.element(Lolium$plantid, whole.sample),]
```

Then, we add the flowering class to the subset data table as a categorical variable by first creating a vector containing all sampled plants with many flowers

```
> manyfl.sample<- append(as.vector(bolu.manyfl.sample),
as.vector(sisu.manyfl.sample) )
> manyfl.sample<- append(as.vector(manyfl.sample),
as.vector(baca.manyfl.sample) )
> manyfl.sample<- append(as.vector(manyfl.sample),
as.vector(bile.manyfl.sample) )
> manyfl.sample<- append(as.vector(manyfl.sample),
as.vector(deru.manyfl.sample) )
> manyfl.sample<- append(as.vector(manyfl.sample),
as.vector(veyo.manyfl.sample) )
```

and then applying factor with `ifelse`

```
> Lolium.sample$flclass<-
factor(ifelse(is.element(Lolium.sample$plantid, manyfl.sample),
"manyfl", "nofl" )
```

It is convenient to create a grouped data table

```
> library(nlme)
> Lolium.sample<- groupedData(cdiam ~ as.numeric(time) | plantid,
outer=~pop*flclass, data= Lolium.sample )
```

which describes the relationship that we want to investigate (`cdiam ~ time`) and the levels of grouping (plant, population, flowering class).

The last step of data preparation is to arrange the levels of population in a sensible sequence,

```
> Lolium.sample$pop<- factor(Lolium.sample$pop, levels=c("Bolu",
"Sisu", "Baca", "Bile", "Deru", "Veyo" ) )
```

with the northern populations (Bolu, Sisu) first, then the central European ones (Baca, Bile), and then the mediterranean ones (Deru, Veyo).

Now we can create the plot

```
> trellis.device(color=F)
> plot(Lolium.sample, key=NULL, xlab="", ylab="Adjusted clone diameter
[cm]", outer=TRUE, aspect=1.5, ylim=c(-8,8), between= list(x=
c(0.5),y= c(0.5)), scales=list(alternating=F, tck= c(1,0),
y=list(at=c(-4, 0, 4)), x=list(tick.number=3)) )
```

that is shown in Fig. 2.

Next we want to explore the relationship between clone diameter and the number of flowering stems, and to see if this differs among populations and genotypes.

We can do that by calculating the regression of $cdiam \sim flowers$ for each genotype and time point. First we create subsets of `Lolium` for each time point

```
> Lolium.t1<- subset(Lolium, time=="t1" )
> Lolium.t2<- subset(Lolium, time=="t2" )
> Lolium.t3<- subset(Lolium, time=="t3" )
```

and then we use the function `lmList` of the *nlme* package for calculating the regressions:

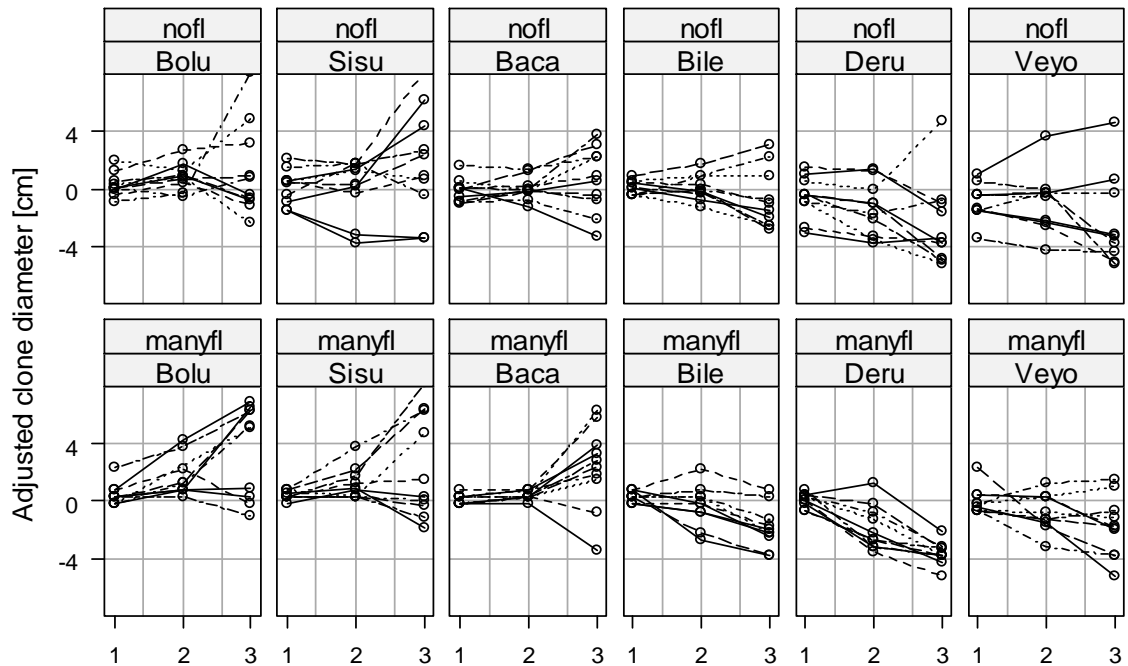


Figure 2 Development of clone diameter (centered within vernalisation treatments) over three time points for random samples of ten plants per population (Bolu, Sisu, ...) and per flowering class (nofl, no flowering stems; manyfl, many (>20) flowering stems at the first time point). The measurements made on the same plant are connected by lines.

```
> Bolu.list.t1<- lmList(cdiam~flowers|gentyp, data=Lolium.t1,
subset=pop=="Bolu" )
> Sisu.list.t1<- lmList(cdiam~flowers|gentyp, data=Lolium.t1,
subset=pop=="Sisu" )
> Baca.list.t1<- lmList(cdiam~flowers|gentyp, data=Lolium.t1,
subset=pop=="Baca" )
> Bile.list.t1<- lmList(cdiam~flowers|gentyp, data=Lolium.t1,
subset=pop=="Bile" )
> Deru.list.t1<- lmList(cdiam~flowers|gentyp, data=Lolium.t1,
subset=pop=="Deru" )
> Veyo.list.t1<- lmList(cdiam~flowers|gentyp, data=Lolium.t1,
subset=pop=="Veyo" )
> Bolu.list.t2<- lmList(cdiam~flowers|gentyp, data=Lolium.t2,
subset=pop=="Bolu" )
> Sisu.list.t2<- lmList(cdiam~flowers|gentyp, data=Lolium.t2,
subset=pop=="Sisu" )
> Baca.list.t2<- lmList(cdiam~flowers|gentyp, data=Lolium.t2,
subset=pop=="Baca" )
> Bile.list.t2<- lmList(cdiam~flowers|gentyp, data=Lolium.t2,
subset=pop=="Bile" )
> Deru.list.t2<- lmList(cdiam~flowers|gentyp, data=Lolium.t2,
subset=pop=="Deru" )
```

```

> Vejo.list.t2<- lmList(cdiam~flowers|genty, data=Lolium.t2,
subset=pop=="Veyo" )
> Bolu.list.t3<- lmList(cdiam~flowers|genty, data=Lolium.t3,
subset=pop=="Bolu" )
> Sisu.list.t3<- lmList(cdiam~flowers|genty, data=Lolium.t3,
subset=pop=="Sisu" )
> Baca.list.t3<- lmList(cdiam~flowers|genty, data=Lolium.t3,
subset=pop=="Baca" )
> Bile.list.t3<- lmList(cdiam~flowers|genty, data=Lolium.t3,
subset=pop=="Bile" )
> Deru.list.t3<- lmList(cdiam~flowers|genty, data=Lolium.t3,
subset=pop=="Deru" )
> Vejo.list.t3<- lmList(cdiam~flowers|genty, data=Lolium.t3,
subset=pop=="Veyo" )

```

Then we extract the slope estimates from the `lmList` objects

```

> Bolu.coef.t1<- coef(Bolu.list.t1)
> Sisu.coef.t1<- coef(Sisu.list.t1)
> Baca.coef.t1<- coef(Baca.list.t1)
> Bile.coef.t1<- coef(Bile.list.t1)
> Deru.coef.t1<- coef(Deru.list.t1)
> Vejo.coef.t1<- coef(Vejo.list.t1)
> Bolu.coef.t2<- coef(Bolu.list.t2)
> Sisu.coef.t2<- coef(Sisu.list.t2)
> Baca.coef.t2<- coef(Baca.list.t2)
> Bile.coef.t2<- coef(Bile.list.t2)
> Deru.coef.t2<- coef(Deru.list.t2)
> Vejo.coef.t2<- coef(Vejo.list.t2)
> Bolu.coef.t3<- coef(Bolu.list.t3)
> Sisu.coef.t3<- coef(Sisu.list.t3)
> Baca.coef.t3<- coef(Baca.list.t3)
> Bile.coef.t3<- coef(Bile.list.t3)
> Deru.coef.t3<- coef(Deru.list.t3)
> Vejo.coef.t3<- coef(Vejo.list.t3)

```

and create boxplots of the estimates for each time point and population.

```

> par(mfrow=c(1,3) )
>
> boxplot(Bolu.coef.t1[,2], Sisu.coef.t1[,2], Baca.coef.t1[,2],
Bile.coef.t1[,2], Deru.coef.t1[,2], Vejo.coef.t1[,2], main="t1",
las=2, names=c("Bolu", "Sisu", "Baca", "Bile", "Deru", "Veyo"),
ylim=c(-0.5, 0.5) )
> boxplot(Bolu.coef.t2[,2], Sisu.coef.t2[,2], Baca.coef.t2[,2],
Bile.coef.t2[,2], Deru.coef.t2[,2], Vejo.coef.t2[,2], main="t2",
las=2, names=c("Bolu", "Sisu", "Baca", "Bile", "Deru", "Veyo"),
ylim=c(-0.5, 0.5) )
> boxplot(Bolu.coef.t3[,2], Sisu.coef.t3[,2], Baca.coef.t3[,2],
Bile.coef.t3[,2], Deru.coef.t3[,2], Vejo.coef.t3[,2], main="t3",
las=2, names=c("Bolu", "Sisu", "Baca", "Bile", "Deru", "Veyo"),
ylim=c(-0.7, 0.7))

```

that are shown in Fig. 3.

Choosing the model setup

What kind of model should we use for this dataset? The dependent variable `cdiam` is continuous metric, so there is some hope that it might be normally distributed. A look at histograms of `cdiam` for the three time points of measurement shows that the

distribution of values is unimodal, but somewhat skewed (Fig. 4). The variance of *cdiam* appears to increase over time. Boxplots of *cdiam* vs. *time* confirm this impression (Fig. 5).

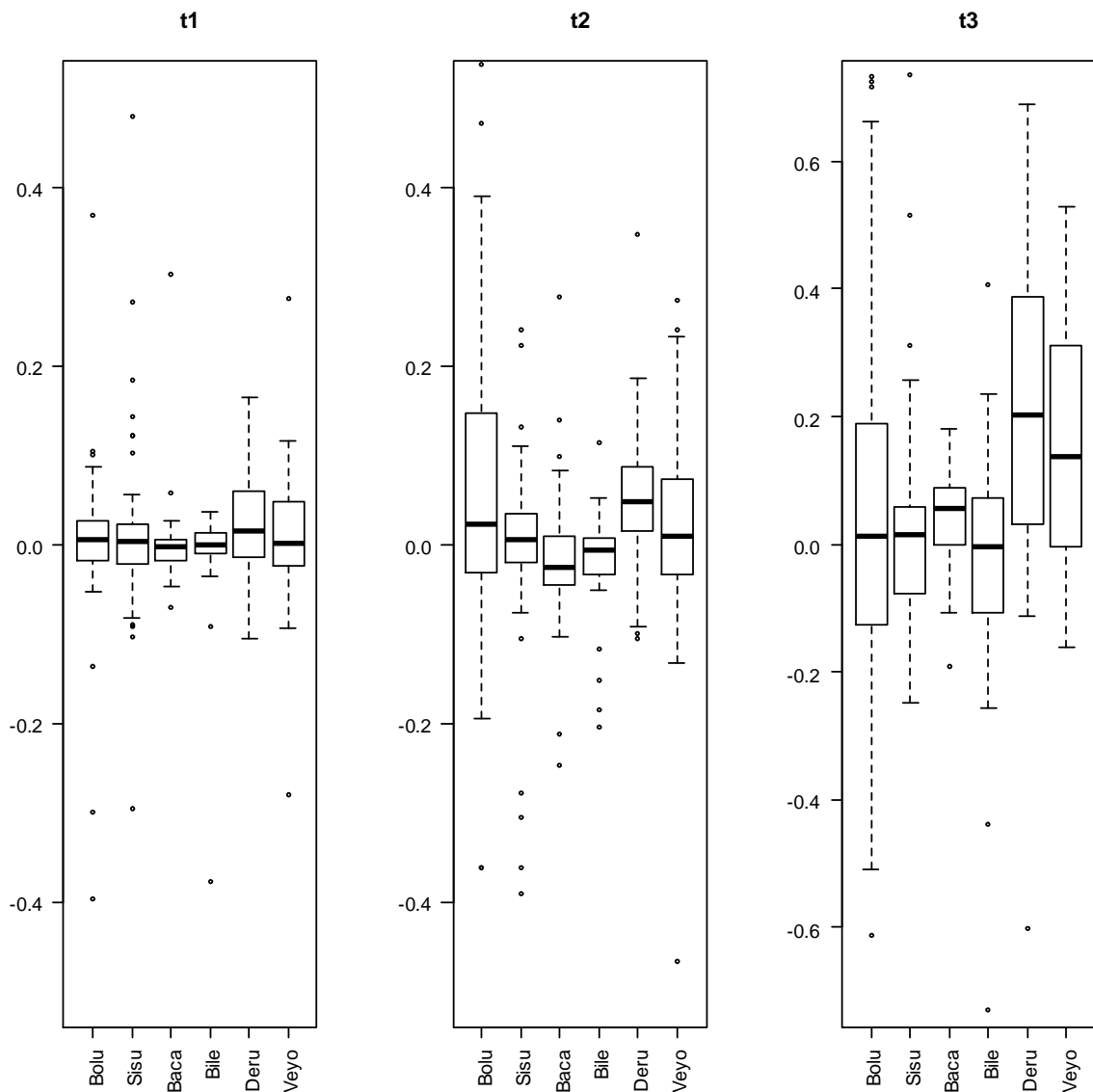


Figure 3 Boxplots of slope estimates of the model "centered clone diameter ~ number of flower stems" calculated for each of 315 genotypes and for three time points. Boxes are drawn for each of six populations.

We could calculate three separate models, one for each time point. Then, the data might fulfill the assumptions of homogeneity of variance. As the design is almost balanced and the potential random grouping variables (population, genotype) are nested, a classical ANOVA setup might be appropriate, then. However, it is more powerful to use the whole dataset in one model. In that case, we have to deal with heterogeneity of variance and with the fact that the data are repeated measures of the same plants that are likely to be auto-correlated. Hence, we will have to use an LMM that can cope with these issues.

Linear mixed effects model with *nlme*

We will use the *lme* function of the R package *nlme* for this analysis. The basic form of the model that we will use to analyse the relationship between number of flowering

stems and diameter of the clones reflects the growth of plants over time: $cdiam \sim b_0 + b_1 \cdot time$.

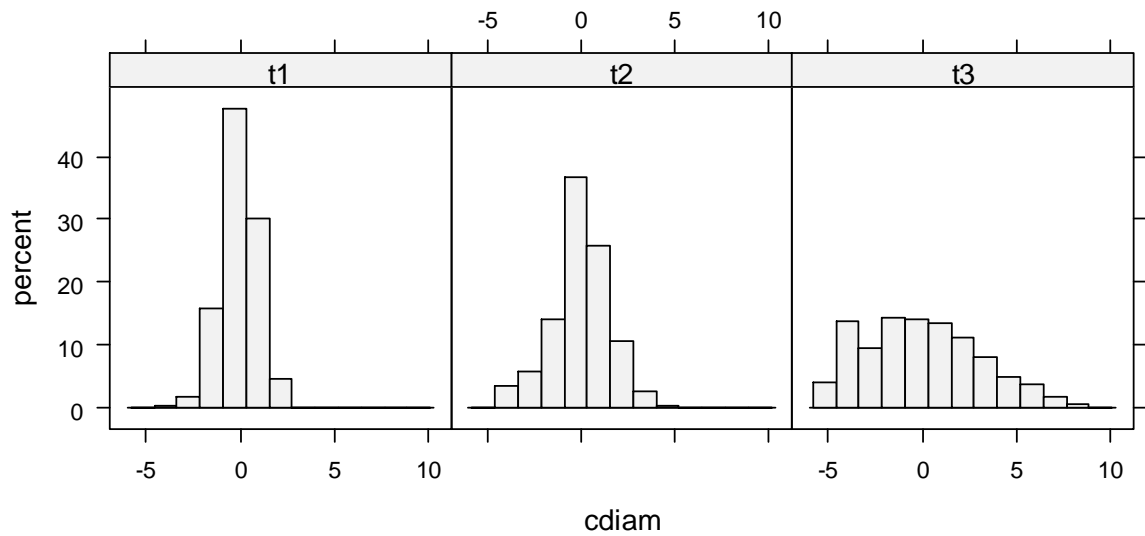


Figure 4 Histograms of centered clone diameter (cdiam) for three time points of measurement (t1-3).

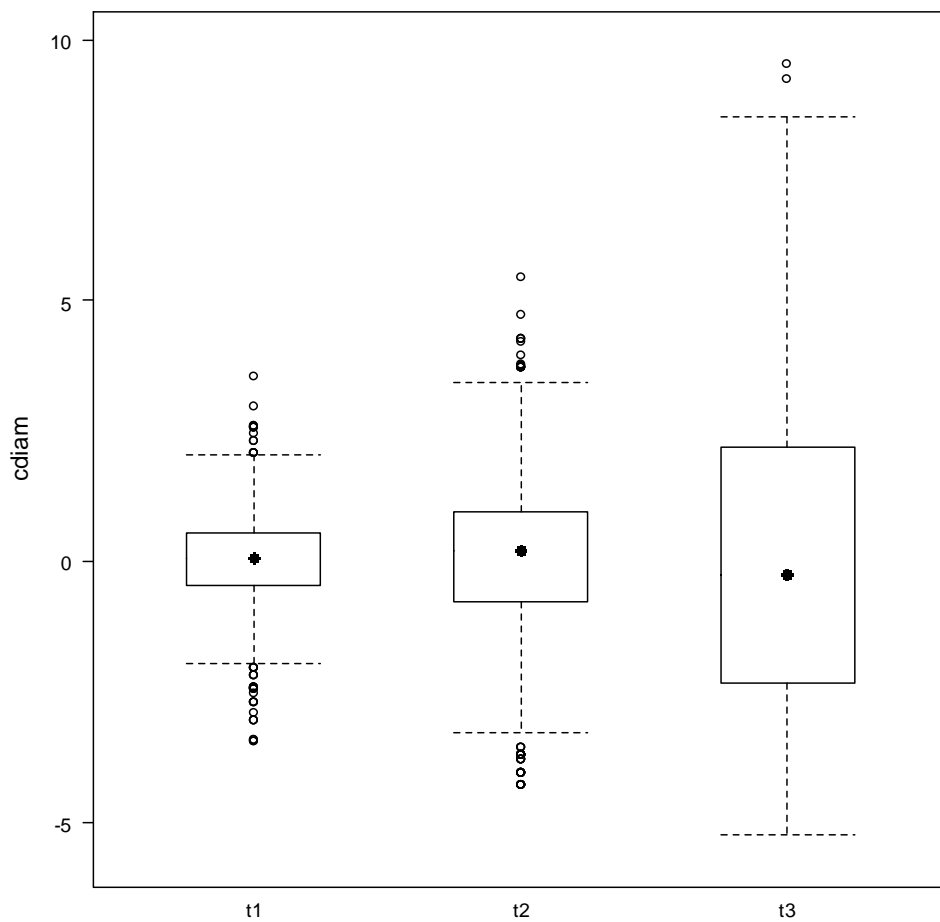


Figure 5 Boxplots of centered clone diameter (cdiam) for three time points of measurement (t1-3).

We hypothesize that the clone diameter at t1 (i.e. b_0) and the growth of the clones after t1 (i.e. b_1) depend on the number of flower stems that the clone produced, the diameter of the competing clone and population. Substituting the coefficients of the

basic model (b_0 , b_1) with `flowers + ccodiam + pop` leads to the following model formula:

```
cdiam ~ flowers + ccodiam + pop + (flowers + ccodiam + pop)*time
```

The main effects of time should be included, too, so the model formula for fixed effects in standard notation can be written as:

```
cdiam ~ flowers*time + ccodiam*time + pop*time
```

Populations are a sample of all possible populations and, thus, might also be treated as a random variable, but in this case, they represent a gradient of vernalisation requirement (northern populations: high; southern populations: low) and there might be meaningful differences among populations. That is why we decided to include population as fixed effect.

Genotype, however, will be included as a random variable because the seeds from which they were grown represent a random sample of all possible seeds within the populations and we are rather interested in the variation among genotypes than in effects of single genotypes. We also include plant (`plantid`) as a random effect nested in genotype (`gentyp`) to account for non-independence of the repeated measures. Fig. 3 suggests that the relationship (the slope) between clone diameter and flowers varies considerably among genotypes. Therefore, we also include a random slope of flowers. In *lme* notation the random effects structure of the model is:

```
random= ~flowers|gentyp/plantid
```

Now we calculate the model outlined above

```
> ml<- lme(cdiam ~ flowers*time + ccodiam*time + pop*time, random=
~flowers|gentyp/plantid, data= Lolium )
```

and have a first look at the results:

```
> summary(ml)
Linear mixed-effects model fit by REML
Data: Lolium
      AIC      BIC    logLik
13834.99 14028.13 -6886.496

Random effects:
Formula: ~flowers | gentyp
Structure: General positive-definite, Log-Cholesky parametrization
           StdDev      Corr
(Intercept) 0.88581455 (Intr)
flowers      0.01727687 -0.685

Formula: ~flowers | plantid %in% gentyp
Structure: General positive-definite, Log-Cholesky parametrization
           StdDev      Corr
(Intercept) 5.180336e-01 (Intr)
flowers      1.608239e-06 0
Residual     1.302583e+00

Fixed effects: cdiam ~ flowers * time + ccodiam * time + pop * time
           Value Std.Error   DF    t-value p-value
(Intercept)  0.128701 0.12369650 2479    1.040455  0.2982
```

flowers	0.009461	0.00316406	2479	2.990303	0.0028
timet2	0.404978	0.10620213	2479	3.813278	0.0001
timet3	0.746519	0.10820030	2479	6.899418	0.0000
ccodiam	0.103919	0.04879531	2479	2.129693	0.0333
popSisu	-0.114216	0.16984848	309	-0.672456	0.5018
popBaca	-0.074247	0.20479563	309	-0.362540	0.7172
popBile	-0.077055	0.20432414	309	-0.377124	0.7063
popDeru	-0.574110	0.20614248	309	-2.785015	0.0057
popVeyo	-0.726351	0.20717091	309	-3.506045	0.0005
flowers:timet2	0.002178	0.00386608	2479	0.563440	0.5732
flowers:timet3	0.024772	0.00420767	2479	5.887286	0.0000
timet2:ccodiam	-0.254712	0.05145968	2479	-4.949740	0.0000
timet3:ccodiam	-0.243609	0.04908139	2479	-4.963361	0.0000
timet2:popSisu	-0.253620	0.14786507	2479	-1.715212	0.0864
timet3:popSisu	-0.077136	0.14915546	2479	-0.517151	0.6051
timet2:popBaca	0.028110	0.18095525	2479	0.155343	0.8766
timet3:popBaca	0.425247	0.18253032	2479	2.329731	0.0199
timet2:popBile	-0.710697	0.18559068	2479	-3.829380	0.0001
timet3:popBile	-2.252353	0.18262202	2479	-12.333414	0.0000
timet2:popDeru	-1.551962	0.19058545	2479	-8.143129	0.0000
timet3:popDeru	-3.298713	0.19398776	2479	-17.004746	0.0000
timet2:popVeyo	-0.706769	0.18526363	2479	-3.814937	0.0001
timet3:popVeyo	-1.841335	0.19059766	2479	-9.660848	0.0000

We plot some diagnostic graphs to evaluate, if the assumptions of normality and homogeneity of variances of the residuals are fulfilled. We create a plot of standardized residuals vs. fitted values, a qqnorm plot, and a plot of standardized residuals vs. time

```
> rf1<- plot(m1)
> qq1<- qqnorm(m1)
> rt1<- plot(m1, resid(.) ~ as.numeric(time)|pop )
```

and, for convenience, we plot the three figures onto one page of a trellis graph

```
> trellis.device(color=F)
> print(rf1, position=c(0,0.5,0.5,1), more=T )
> print(qq1, position=c(0.5,0.5,1,1), more=T )
> print(rt1, position=c(0,0,1,0.5) )
```

The plots shown in Fig. 6 suggest that the distribution of the residuals is not that far away from normality, but indeed the scatter of residuals increases with time.

Therefore, we will try to solve this problem by including a variance structure into the model using the `weights` argument with a `varIdent` formula that allows for separate variance estimates for each time point. For convenience, we use `update`

```
> m2<- update(m1, weights= varIdent(form= ~1|time) )
```

but get an error message saying that the model did not converge. We try some simpler random effects structures and find out that a model with random effects of `plantid`, but without `gentyp`, converges. This random effect structure is not the perfect choice, but we suppose that it should capture much of the variation among genotypes, too, and it allows accounting for correlation of the measurements. So we continue the analysis with the simpler random structure and calculate the model with separate variance estimates for each time point

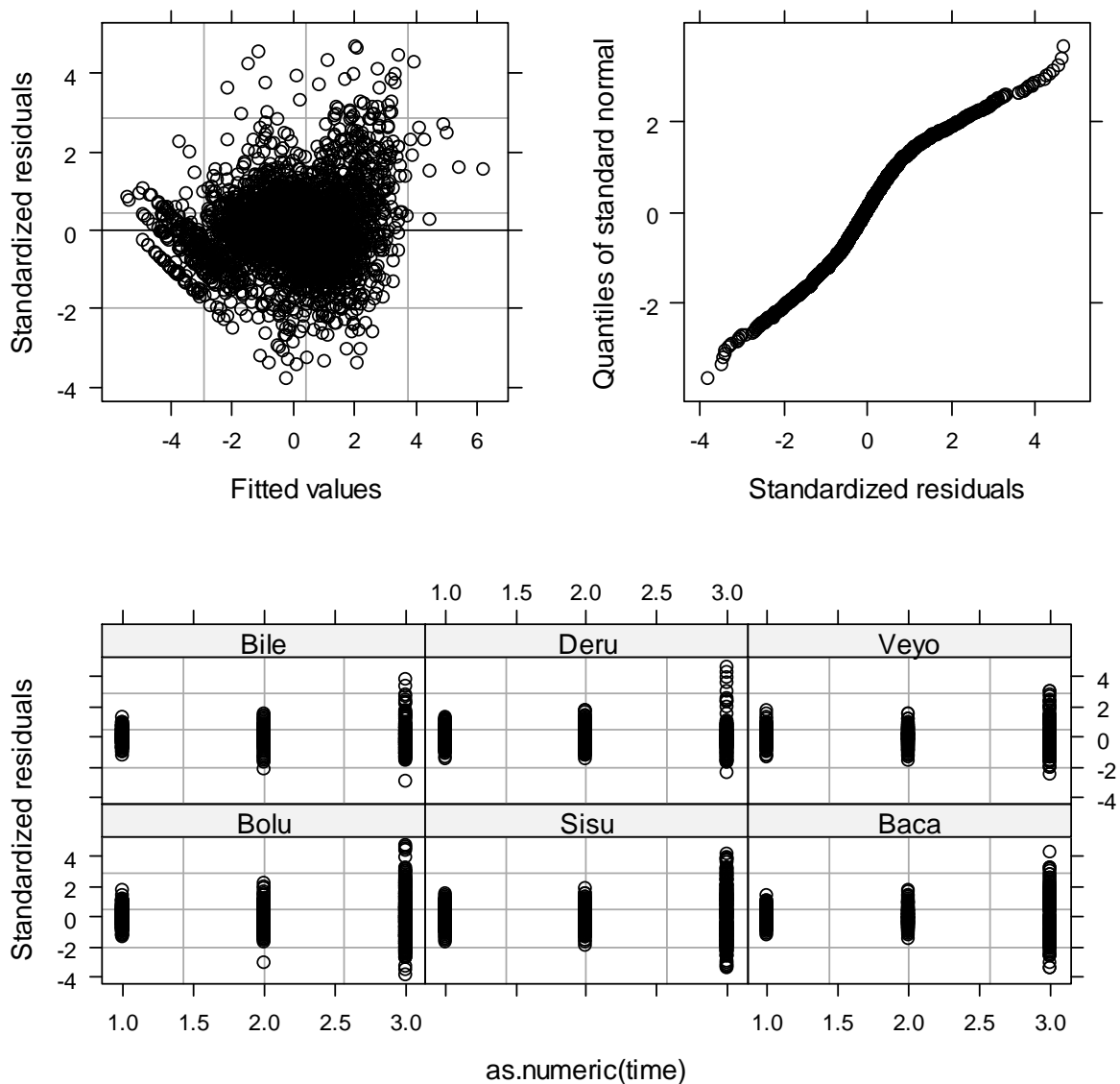


Figure 6 Diagnostic plots of model m1.

```
> m2<- lme(cdiam ~ flowers*time + ccodiam*time + pop*time, random=
~flowers|plantid, data= Lolium, weights= varIdent(form= ~1|time) )
```

Then, we draw the diagnostic plots again

```
> trellis.device(color=F)
> rf2<- plot(m2)
> qq2<- qqnorm(m2)
> rt2<- plot(m2, resid(., type="p") ~ as.numeric(time)|pop )
> print(rf2, position=c(0,0.5,0.5,1), more=T )
> print(qq2, position=c(0.5,0.5,1,1), more=T )
> print(rt2, position=c(0,0,1,0.5) )
```

and see that the qqnorm plot suggests improved normality of residuals and that the variation of residuals does not increase with time anymore (Fig. 7).

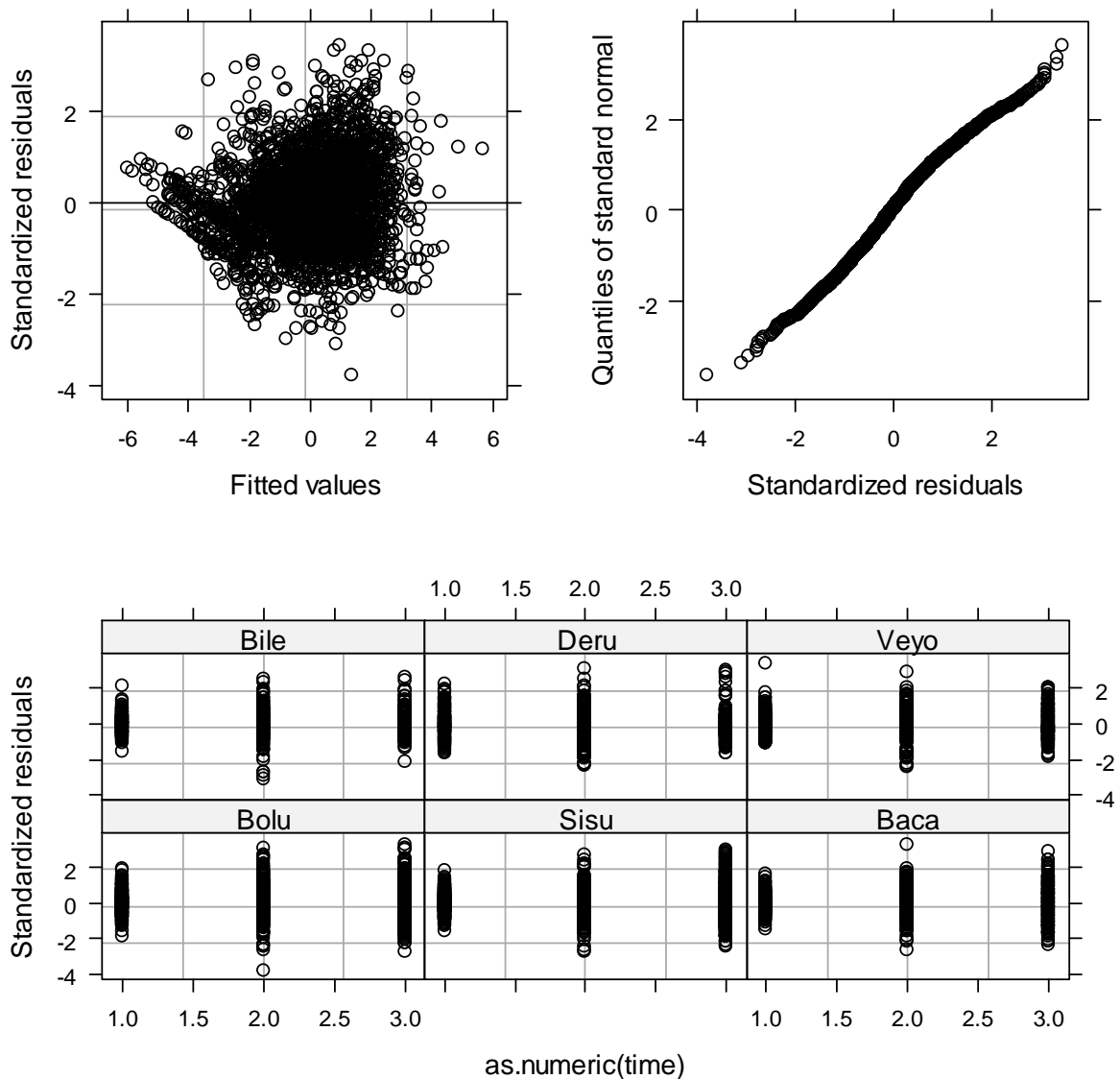


Figure 7 Diagnostic plots of model m2.

The last step towards a valid model structure will be to implement an auto-correlation structure of measurements made on the same plants. Until now we have included a random intercept and slope for `plantid`, but we have not taken into account that measurements made at consecutive time points may be more correlated than measurements further apart.

To do so, we will define a first-order auto-correlation structure using the `corAR1` argument

```
> m3<- update(m2, correlation= corAR1(form= ~1|plantid) )
```

and test whether this model (m3) fits better than the model with unstructured correlation of residuals within plant individuals (m1 and m2). We can test this with `anova` that performs a likelihood ratio tests comparing m2 and m3

```
> anova(m2, m3)
  Model df      AIC      BIC    logLik  Test  L.Ratio p-value
m2     1  30 12175.49 12362.40 -6057.746
m3     2  31 12037.02 12230.16 -5987.509 1 vs 2 140.4734 <.0001
```

and we see that the auto-correlation structure significantly improved the model fit.

Now that we have built a random effects structure that suits the data, we can go ahead with testing the fixed effects. As REML estimates are not appropriate for fixed effects, we recalculate `m3` using ML.

```
> m3ML<- update(m3, method = "ML" )
```

and then we conduct Wald F tests of the fixed effects (when used on single lme objects, `anova` produces Wald F tests)

```
> anova(m3ML)
              numDF denDF  F-value p-value
(Intercept)      1  2479   2.3222  0.1277
flowers           1  2479 112.6673 <.0001
time              2  2479   0.3921  0.6757
ccodiam           1  2479 364.0616 <.0001
pop               5  1274  29.2452 <.0001
flowers:time      2  2479  14.0307 <.0001
time:ccodiam     2  2479  35.8031 <.0001
time:pop         10  2479  51.9991 <.0001
```

and see that there is a significant relationship between clone diameter and number of flower stems. But in the summary table

Linear mixed-effects model fit by maximum likelihood

Data: Lolium

	AIC	BIC	logLik
	11931.15	12124.48	-5934.574

Random effects:

Formula: ~flowers | plantid

Structure: General positive-definite, Log-Cholesky parametrization

	StdDev	Corr
(Intercept)	0.68560889	(Intr)
flowers	0.01638808	-0.924
Residual	0.54497651	

Correlation Structure: AR(1)

Formula: ~1 | plantid

Parameter estimate(s):

Phi

0.4101318

Variance function:

Structure: Different standard deviations per stratum

Formula: ~1 | time

Parameter estimates:

	t1	t2	t3
	1.000000	1.840848	4.234451

Fixed effects: `cdiam ~ flowers * time + ccodiam * time + pop * time`

	Value	Std.Error	DF	t-value	p-value
(Intercept)	0.133666	0.04718863	2479	2.832596	0.0047
flowers	0.011189	0.00133687	2479	8.369552	0.0000
timet2	0.408767	0.05385807	2479	7.589713	0.0000
timet3	0.799756	0.13454029	2479	5.944364	0.0000
ccodiam	-0.121047	0.02246991	2479	-5.387053	0.0000
popSisu	-0.168501	0.06441466	1274	-2.615874	0.0090
popBaca	-0.178257	0.07849019	1274	-2.271074	0.0233
popBile	-0.098700	0.07735102	1274	-1.276002	0.2022
popDeru	-0.493595	0.07657919	1274	-6.445550	0.0000
popVeyo	-0.775559	0.07781317	1274	-9.966940	0.0000

flowers:timet2	0.001864	0.00193511	2479	0.963024	0.3356
flowers:timet3	0.020360	0.00466267	2479	4.366502	0.0000
timet2:ccodiam	-0.083359	0.02324034	2479	-3.586823	0.0003
timet3:ccodiam	-0.022484	0.03061635	2479	-0.734371	0.4628
timet2:popSisu	-0.226873	0.07524390	2479	-3.015171	0.0026
timet3:popSisu	-0.040378	0.18504946	2479	-0.218203	0.8273
timet2:popBaca	0.040763	0.09204452	2479	0.442856	0.6579
timet3:popBaca	0.429529	0.22690483	2479	1.892990	0.0585
timet2:popBile	-0.697108	0.09545838	2479	-7.302747	0.0000
timet3:popBile	-2.219989	0.22682941	2479	-9.787043	0.0000
timet2:popDeru	-1.581617	0.09756315	2479	-16.211212	0.0000
timet3:popDeru	-3.453770	0.23771008	2479	-14.529338	0.0000
timet2:popVeyo	-0.716729	0.09465597	2479	-7.571938	0.0000
timet3:popVeyo	-1.928666	0.23984115	2479	-8.041431	0.0000

the estimates for `flowers` and `flowers:time` show that this relationship is positive. Thus, we conclude that there is co-occurrence of the two reproduction strategies instead of trade-off as hypothesized.

References

- Fox J. Linear mixed models. Appendix to *An R and S-PLUS companion to Applied Regression*. Available from: URL: <http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-mixed-models.pdf>; 2002:
- Pinheiro J, Bates D, DebRoy S, Sarkar D, R Development Core Team (2011). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-98.
- R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- Sarkar D (2008) Lattice: Multivariate Data Visualization with R. Springer, New York. ISBN 978-0-387-75968-5.
- Thiele J, Jørgensen RB, Hauser T. Flowering does not decrease vegetative competitiveness of *Lolium perenne*. *Basic and Applied Ecology* 2009;10:340–8.