New Phytologist Supporting Information

Tansley Insight

How to analyse plant phenotypic plasticity in response to a changing climate

Pieter A. Arnold, Loeske E. B. Kruuk, Adrienne B. Nicotra

Supporting Information Notes S1

Common basis functions for non-linear reaction norms and software resources for fitting random regression mixed models

Here, we provide a brief overview of some common basis functions that could be used to describe the population-level reaction norms of various shapes in random regression mixed models (RRMMs) and resources for how to implement these models. We have already presented linear and quadratic functions in the fixed effects component of equations (2) and (3) in Box 2, so we continue on from there.

Polynomial functions are expandable to the nth-degree to describe complex curves:

$$y_i = \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_n x_i^n \quad (4)$$

where y_i is the phenotypic trait on occasion i, α and β_n are the intercept and slope coefficients, respectively, x_i is the environment that was exposed to on occasion i, and the largest exponent n of x_i^n determines the number of slope coefficients and the degree of the polynomial function. Higher-degree polynomial functions may be useful for modelling reaction norms that have peaked distributions, thresholds, or longer tails than what can be effectively modelled by a quadratic function (e.g., Petavy *et al.*, 2001). Alternate forms of the basic polynomial function, such as orthogonal Legendre polynomials, may be useful as functions for reaction norms with continuous environments and growth trajectories over time (e.g., Kirkpatrick *et al.*, 1990; Stratton, 1998; Marchal *et al.*, 2018).

The logistic function underlies reaction norms that have a sigmoidal (or S-) shape, which are defined by the function:

$$y_i = \frac{L}{1+e^{-(\alpha+\beta x_i)}} \quad (5)$$

where the coefficients are as above, with additional terms L, the maximum value of the curve, and e, the natural logarithm base. Specific logistic functions, such as Gompertz function, have long been applied to growth data that are non-symmetrical (Richards, 1959; Paine *et al.*, 2011) and this family of

functions may prove useful for analysis of phenological or growth data that match the properties of these functions (e.g., Jochner *et al.*, 2016).

Any mathematical function can theoretically be applied to model the population-level response (Morrissey & Liefting, 2016), however there is always a trade-off between the adequacy of the mathematical function and biological pragmatism (and subsequent interpretability). One such function that could be explored for complex population-level reaction norm shapes is a generalised additive model (GAM), which builds a smoothed spline from multiple polynomial regressions pieced together (Wood, 2017). The general form is:

$$y_i = \alpha + f(x_i) \quad (6)$$

where $f(x_i)$ is a smoothing spline function either with a specific parameter form (e.g., a cubic polynomial), or may be a non-parametric smoothing parameter (e.g., estimated from locally weighted means). The benefit of this approach is that the function is less constrained by an *a priori* shape, and is better informed by the data itself. Conversely, one must take care not to over-parameterise GAMs if the study design has relatively few measurement points along the environmental axis.

Selecting the appropriate basis function to model the population average reaction norm should be based on the data itself, rather than prior reaction norm shape assumptions. First, a series of different functions that could potentially describe the mean population-level reaction norm should be fitted. Then, standard model selection approaches (e.g., Akaike Information Criterion (AIC), likelihood-ratio-tests, and backward stepwise selection) can be used to determine the appropriate balance between the fit of the function to the data and the complexity that will need to be interpreted once included in a mixed model. For a very useful introduction to mixed models, and discussion of the conceptual difference between fixed and random effects, see Chapter 19 in Crawley (2012). For further reading on model selection and fitting the RRMMs that we have outlined, Zuur *et al.* (2009) is a comprehensive resource.

There are some cautionary points about the RRMM approach that may be worth considering. First, to confidently estimate the variance among individual reaction norms, RRMMs – particularly models with additional covariates and complex non-linear random components – might require more data than typically collected in plasticity studies to achieve adequate statistical power (see Martin *et al.*, 2011; van de Pol, 2012 for suggested rules-of-thumb, power analyses, and other considerations for study design). Second, the shape of estimated non-linear functions may be affected by the distribution of environments over which the phenotype was measured and may not be a function specifically of the biology of the system (Morrissey & Liefting, 2016). This issue can be generally avoided by ensuring that the data reasonably match the fitted function by assessing model fit

diagnostics. Third, under skewed data distributions, linear coefficient estimates may be biased by the inclusion of non-linear terms in the model. The magnitude of this bias effect can be reduced if the temperature (or other environmental measure) is mean-centred overall and the within-individual reaction norms are also mean-centred prior to analysis, which is particularly important if the environmental range differs between individuals (van de Pol & Wright, 2009).

Various software packages are available for `R` (R Development Core Team, 2017) in which RRMMs can be fitted, including `lme4` (Bates *et al.*, 2015), `nlme` (Pinheiro *et al.*, 2014), `MCMCglmm` (Hadfield, 2010), `mgcv` (Wood, 2017), and `ASReml-R` (Butler *et al.*, 2009). Additional packages such as `pamm` (Martin *et al.*, 2011) and `MuMIn` (Johnson, 2014) include functions to conduct power analyses and evaluate goodness-of-fit for RRMMs. Together with their extensive resource libraries, these software packages should allow RRMMs to be readily implemented in plant plasticity research, such as in the worked example we provide in Supporting Information Notes S2.

Supporting Information Notes S2

Worked example of random regression mixed models

Here, we provide a tutorial to demonstrate the use of random regression mixed models (RRMMs) for a plant phenotypic plasticity dataset, using the example .csv data file provided in Supporting Information Notes S3 and R code file in Supporting Information Notes S4. This worked example will guide the user through the process of developing models from simple linear regression without any random effects, to more complex mixed models with different basis functions for both fixed and random effects. The purpose of this tutorial is primarily to demonstrate how to fit these models in R, but not to cover the underlying statistical theory or interpretation of results. We assume some basic understanding of linear modelling and R code, but attempt to provide accessible and reproducible R code for implementing these models in the lme4 package that can be applied to a variety of datasets with modification.

Scenario

Imagine we are interested in 1) mapping the response of Spring flowering time advancement to increased temperatures in a cold-climate plant species and 2) determining the variation among a series of bred genotypes of this species. A simplistic experimental design aimed to address this question of whether growth temperature affects phenology might be as follows. First, 10 individuals of each of 20 genotypes of a plant species are grown under common glasshouse conditions, then once at a suitable growth stage, one individual from each genotype is transferred into one of ten controlled temperature growth cabinets ranging from 5 to 20°C, and the Julian day that the plant produces its first flower is recorded. For simplicity, the Julian day is scaled so that it is centred on a known relative date where the plants flower under typically cool conditions. Thus, when the plants are exposed to increasingly warm growth temperatures, the relative date of flowering becomes more negative. We note that the data that we discuss here are simulated and are modelled without additional covariates that might be present in a more complex experimental design, therefore the model specifications and fits may be more ideal and simplistic than real biological data. However, the principles remain the same and we hope that this will be a useful exercise in the philosophy of building and applying random regression mixed models.

Download, install, or update R packages

Packages can be downloaded through R or RStudio directly, but links are provided for all packages so that users can access additional materials such as vignettes, if desired.

R: https://cran.r-project.org/

RStudio: https://www.rstudio.com/

ggplot2: https://cran.r-project.org/web/packages/ggplot2/index.html

MuMIn: https://cran.r-project.org/web/packages/MuMIn/index.html

lme4: https://cran.r-project.org/web/packages/lme4/index.html

- Note that the `r.squaredGLMM` function requires MuMIn v1.41.0 or later.
- To update R packages within R use the function `update.packages()` before loading the packages that are to be updated into the R session.
- At the time of developing this tutorial, the authors used the following versions:
    - R v3.5.1
    - ggplot2 v3.0.0
    - MuMIn v1.42.1
    - lme4 v1.1-18-1

Acknowledgements

We thank Oliver Binks, Timothée Bonnet, Alexandra Catling, and Shuo Wang for constructive feedback on this tutorial and for testing the R code across Windows, Mac, and Linux operating systems.

Loading libraries and reading data into R

To begin analysing these data, first install (see previous page) and then load in the relevant R libraries.

```
> # Load in libraries
> library(ggplot2)
> library(MuMIn)
> library(lme4)
> # If lmer model p-values are desired then 'lmerTest' may be loaded in place of 'lme4'
>
```

Read the data file into R (the 'flowerdata.csv' file can be downloaded from Supplementary Notes S3) and check the structure of the data.

```
> # Read in data
> flowerdata <- read.csv(file = "flowerdata.csv", header = TRUE, skip = 10)
> head(flowerdata)
  genotype relativedate temperature ID
1        1     1.104052    5.000000  1
2        1     1.063095    6.666667  2
3        1    -1.226558    8.333333  3
4        1    -3.162186   10.000000  4
5        1    -2.516623   11.666667  5
6        1    -2.849256   13.333333  6
> str(flowerdata)
'data.frame': 200 obs. of  4 variables:
 $ genotype    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ relativedate: num  1.1 1.06 -1.23 -3.16 -2.52 ...
 $ temperature : num  5 6.67 8.33 10 11.67 ...
 $ ID          : int  1 2 3 4 5 6 7 8 9 10 ...
>
```

Note that 'genotype' is an integer by default but should be converted to be a factor. Here we also add in another column called 'loc' to assign a location based on temperatures to allow for the same genotypes being represented across 10 temperatures (essentially accounting for repeated measures).

```
> # Convert genotype from integer to factor and check structure again
> flowerdata$genotype <- as.factor(flowerdata$genotype)
> flowerdata$loc <- as.factor(flowerdata$temperature)
> str(flowerdata)
'data.frame': 200 obs. of  4 variables:
 $ genotype    : Factor w/ 20 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ relativedate: num  1.1 1.06 -1.23 -3.16 -2.52 ...
 $ temperature : num  5 6.67 8.33 10 11.67 ...
 $ ID          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ loc         : Factor w/ 10 levels "5","6.666666667",..: 1 2 3 4 5 6 7 8 9 10 ...
>
```

Add in a column of mean-centre temperatures using the `scale` function. Mean-centring of the environmental variable means that intercepts reflect average values for the population and individuals (see Box 2).

```
> # Mean-centre the x variable (temperature)
> # From here on, use the mean-centred temperature (ctemperature)
> flowerdata$ctemperature <- scale(flowerdata$temperature)
> head(flowerdata)
genotype relativedate temperature ID         loc ctemperature
1        1     1.104052    5.000000  1           5   -1.5627772
2        1     1.063095    6.666667  2 6.666666667   -1.2154934
3        1    -1.226558    8.333333  3 8.333333333   -0.8682096
4        1    -3.162186   10.000000  4          10   -0.5209257
5        1    -2.516623   11.666667  5 11.66666667   -0.1736419
6        1    -2.849256   13.333333  6 13.33333333    0.1736419
>
```

The first step in analysing these data is to determine the appropriate basis function for the population-average reaction norms. To do so, let us plot the raw values of relative date of flowering against mean-centred growth temperature for each genotype.

```
> # Plot the main effects
> ggplot(flowerdata, aes(x = ctemperature, y = relativedate, group = genotype)) +
+   geom_line(aes(colour = genotype)) + ylab("Relative date of flowering") +
+   xlab("Mean-cented growth temperature") + theme_classic()
>
```

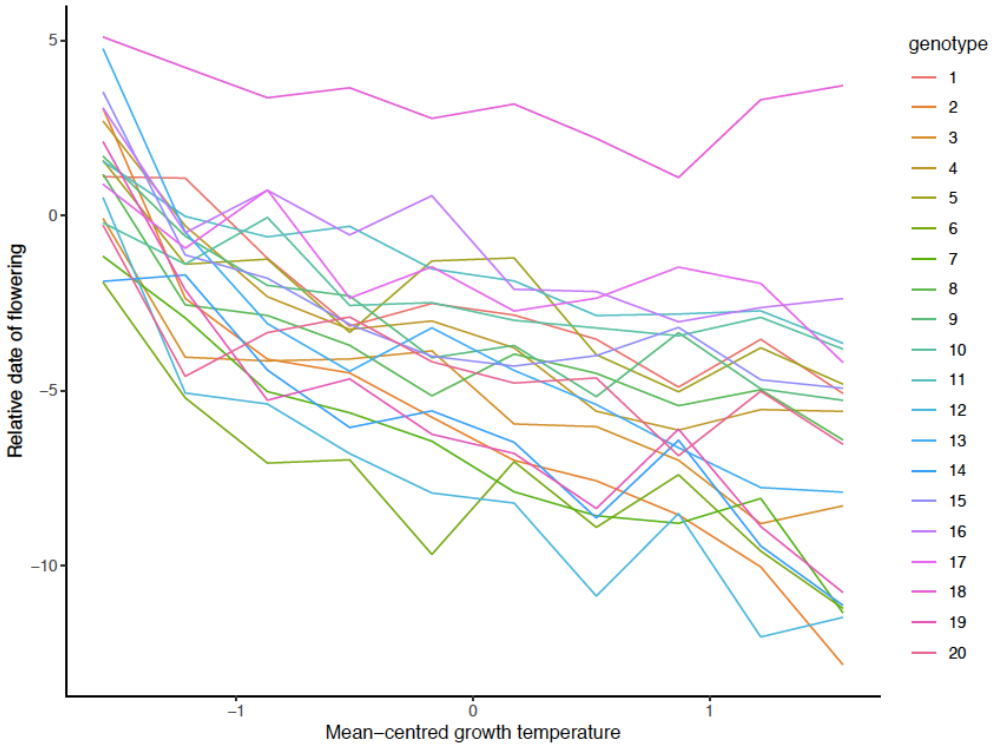

Figure A. Plot of raw data of relative date of flowering across mean-centred growth temperatures for 20 genotypes.

We can observe that there is a general negative trend in the relative date of flowering as growth temperature increases, but also that variance in the relative date of flowering is greater at 20°C compared to 5°C and there appears to be some non-linearity across genotypes (Fig. A). It is therefore worth investigating whether the fixed effect of growth temperature on the relative date of flowering could be modelled with either a linear model or a quadratic model.

Building mixed-effects models

Let us first fit a simple linear mixed model and observe the average population-level reaction norm. Note that checking model diagnostics and residuals for patterns in the data that violate modelling assumptions (and if appropriate, applying transformations or scaling) is standard practice in biology that we assume the user is familiar with and would employ to evaluate model assumptions. For brevity, we will not delve further into model diagnostics here as they are discussed elsewhere for mixed-effects models (Bates *et al.*, 2015).

Here, we add in the random effect (intercept) of `(1|loc)` to account for spatial differences among the ten growth chambers across which each genotype is represented. Further, we specify that the mixed model is to be fitted using maximum likelihood (ML) rather than restricted maximum likelihood (REML) with the term `REML = FALSE` so that models that contain different fixed effects can be compared directly (Zuur *et al.*, 2009).

```
#### Basic linear mixed model ####
# Fit a linear model for the fixed effect of growth temperature on relative date of
flowering.
# Note that we also add a random effect for 'location' to take account of the
# repeated measures at each temperature

> model1.1 <- lmer(relativedate ~ ctemperature + (1|loc), REML = FALSE, data =
flowerdata)

> # Check model summary and R squared values
> summary(model1.1)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: relativedate ~ ctemperature + (1 | loc)
   Data: flowerdata

     AIC      BIC   logLik deviance df.resid
   998.2   1011.4   -495.1    990.2      196

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.1925 -0.6741  0.0005  0.6485  3.7194

Random effects:
 Groups   Name        Variance Std.Dev.
 loc      (Intercept) 0.1062   0.3259
 Residual             8.1796   2.8600
Number of obs: 200, groups:  loc, 10

Fixed effects:
             Estimate Std. Error t value
(Intercept)   -3.6897     0.2270 -16.255
```

```
ctemperature  -2.1124     0.2276  -9.283

Correlation of Fixed Effects:
            (Intr)
ctemperatur 0.000
>
```

We can evaluate the model fit with an R^2 function from the MuMIn package (Bartoń, 2018), called `r.squaredGLMM`, which has been specifically designed for calculating model fit of linear mixed-effects models with a revised statistic (Nakagawa & Schielzeth, 2013; Johnson, 2014). The first time that `r.squaredGLMM` is called, there will be a warning message about this revision.

```
> r.squaredGLMM(model1.1)
           R2m       R2c
[1,] 0.3500338 0.3583672
Warning message:
'r.squaredGLMM' now calculates a revised statistic. See the help page.
>
```

The `R2m` value is the marginal R^2 that is the fit of the fixed-effects only, and the `R2c` value is the conditional R^2 that explains the proportion of variance accounted for by the random and fixed-effects combined. We are therefore interested primarily in the value of `R2c`. The R^2 value of model1.1 is 0.36, which we can use to compare against subsequent models. We can then visually assess how well the linear model fits the raw data by overlaying the regression line from model1.1 as an average population-level reaction norm. To do this we can use the `predict` function to predict y-values across the continuous x-axis, and then plot the fixed effect of temperature from model1.1 over the raw genotype-specific reaction norms (Fig. B).

```
> # Predict values based on the model fit using the predict function
> temperature_pred <- data.frame(ctemperature = seq(from =
min(flowerdata$ctemperature), to = max(flowerdata$ctemperature), length.out = 50))
> temperature_pred$fit1.1 <- predict(model1.1, newdata = temperature_pred, re.form =
NA)
> # Plot the raw data and overlay the fit of Model1.1
> ggplot(temperature_pred, aes(x = ctemperature, y = fit1.1)) +
+   geom_line(data = flowerdata, aes(y = relativedate, colour = genotype)) +
+   geom_line(size = 2) +
+   ylab("Relative date of flowering") + xlab("Mean-cented growth temperature") +
+   theme_classic()
>
```

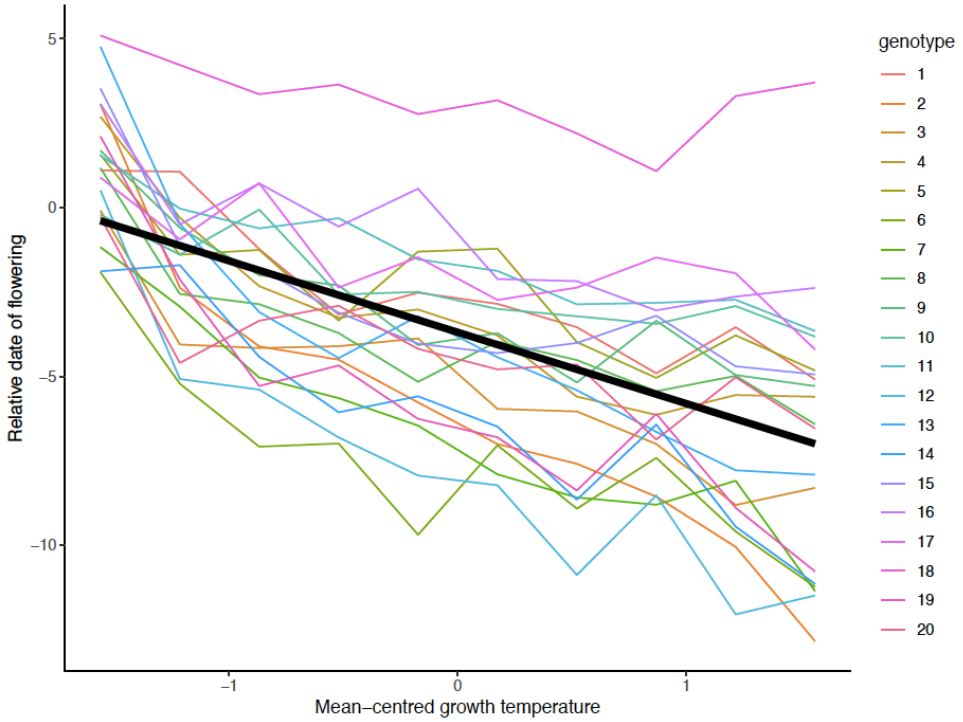

Figure B. Plot of raw data of relative date of flowering across mean-centred growth temperatures for 20 genotypes. The thick black line represents the linear regression model fit (model1.1) of the overall effect of mean-centred growth temperature (the predicted average population-level reaction norm).

The average population-level reaction norm is a fair fit to the data (Fig. B), but note that most of the data at the lowest growth temperatures are above the fitted regression line. This observation suggests that these flowering phenology data may not have a strictly linear relationship with growth temperature and thus could be modelled with a low-order polynomial function (quadratic) to see if that improves model fit.

So, let us fit a quadratic fixed-effect of temperature to model to these data. The degree of polynomial is set by the `poly()` function where degree 2 is a quadratic, 3 is a cubic, and so on.
`poly(ctemperature, 2, raw = T)1` is the slope coefficient of the quadratic function and `poly(ctemperature, 2, raw = T)2` is the quadratic coefficient of the quadratic function that reflects the curvature of the regression.

```
> #### Quadratic fixed effects model ####
> # Fit a quadratic model for the fixed effect of growth temperature on relative date
of flowering
> model1.2 <- lmer(relativedate ~ poly(ctemperature, 2, raw = T) + (1|loc),
+                  REML = FALSE, data = flowerdata)
> # Check model summary and R squared values
> summary(model1.2)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: relativedate ~ poly(ctemperature, 2, raw = T) + (1 | loc)
```

```
   Data: flowerdata

     AIC      BIC   logLik deviance df.resid
   993.9   1010.4   -492.0    983.9      195

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3650 -0.6812  0.0104  0.6009  3.4764

Random effects:
 Groups   Name        Variance Std.Dev.
 loc      (Intercept) 0.000    0.000
 Residual             8.019    2.832
Number of obs: 200, groups:  loc, 10

Fixed effects:
                               Estimate Std. Error t value
(Intercept)                     -4.2757     0.3030 -14.113
poly(ctemperature, 2, raw = T)1  -2.1124     0.2007 -10.523
poly(ctemperature, 2, raw = T)2   0.5890     0.2285   2.578

Correlation of Fixed Effects:
            (Intr) p(,2,r=T)1
pl(,2,r=T)1  0.000
pl(,2,r=T)2 -0.750  0.000
> r.squaredGLMM(model1.2)
           R2m       R2c
[1,] 0.3710001 0.3710001
>
```

The R^2 value of model1.2 is 0.37, which is a marginal improvement in model fit over the linear model1.1. We can also see that the slope of the quadratic model fit is significant and strongly negative and the quadratic (curvature) of the model is much weaker than the slope, but significant and positive (increasingly positive y-values at extreme x-values). What does the average population-level reaction norm modelled by the quadratic regression look like when overlaid over the raw data? We can again use the `predict` function to predict y-values across the continuous x-axis for the non-linear quadratic function, and then plot the quadratic function of model1.2 over the raw genotype-specific reaction norms (Fig. C).

```
> # Predict values based on the model fit using the predict function
> temperature_pred$fit1.2 <- predict(model1.2, newdata = temperature_pred, re.form =
NA)
> # Plot the overall model fit over the top of the raw data
> ggplot(temperature_pred, aes(x = ctemperature, y = fit1.2)) +
+   geom_line(data = flowerdata, aes(y = relativedate, colour = genotype)) +
```

```
+    geom_line(size = 2) +
+    ylab("Relative date of flowering") + xlab("Mean-centred growth temperature") +
+    theme_classic()
>
```

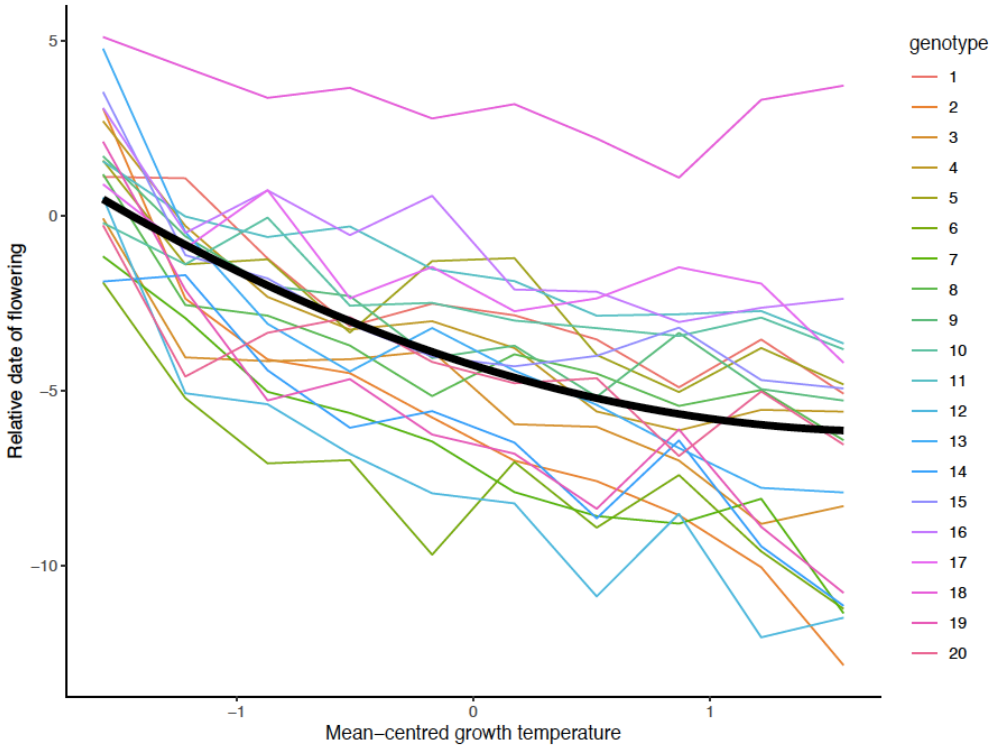

Figure C. Plot of raw data of relative date of flowering across mean-centred growth temperatures for 20 genotypes. The thick black line represents the quadratic regression model fit (model1.2) of the overall effect of mean-centred growth temperature (the predicted average population-level reaction norm).

We can now compare the two models using two methods. The first is to compare the log-likelihoods of the two models and perform a likelihood ratio test (LRT). Here, we take 2 × the log-likelihood values from the model summaries to determine a χ^2 value, and then determine the *p*-value of the LRT from the χ^2 value and the difference of the degrees of freedom (df) between the two models using the function `1-pchisq(chi2, df)`. In this model comparison, there is a df difference of 1.

```
> # Are the two models different?
> # Likelihood ratio test
> chi2 <- 2*(summary(model1.2)$logLik - summary(model1.1)$logLik)
> 1-pchisq(chi2,1)
'log Lik.' 0.0123132 (df=5)
>
```

The resulting *p*-value is 0.012, so the models would be considered significantly different with a typical α threshold of 0.05. We can also then balance the goodness-of-fit against the complexity of the

models by comparing Akaike Information Criterion (AIC) values to help guide the model selection process. The lower the AIC value, the more parsimonious the fit of the model is to the data.

```
> # AIC comparison
> AIC(model1.1, model1.2)
         df      AIC
model1.1  4 998.2138
model1.2  5 993.9486
>
```

Thus, model1.2 has an improved fit to the data and is more parsimonious than model1.1. Hereafter, we will continue to develop the quadratic fixed-effects model (model1.2) with various random effects of interest and compare model fits.

To further develop the model, the overall population-level reaction norm as model1.2 is kept the same, but with an additional term of `(1|genotype)`, which specifies that we are allowing the y-intercept value to vary among genotypes. Hence, model1.3 is a random intercepts-only linear mixed-effects model.

```
> #### Quadratic fixed effects with random intercepts model ####
> # Fit a linear mixed effects model (random intercepts only) for the fixed effect of
> # growth temperature on relative date of flowering and random effect of genotype
intercepts
> model1.3 <- lmer(relativedate ~ poly(ctemperature, 2, raw = T) + (1|loc) +
(1|genotype),
+                  REML = FALSE, data = flowerdata)
> # Check model summary and R squared values
> summary(model1.3)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: relativedate ~ poly(ctemperature, 2, raw = T) + (1 | loc) + (1 |
genotype)
   Data: flowerdata

     AIC      BIC   logLik deviance df.resid
   757.9    777.7   -373.0    745.9      194

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2264 -0.5963  0.0544  0.5306  3.3112

Random effects:
 Groups   Name        Variance Std.Dev.
 genotype (Intercept) 6.2667   2.5033
 loc      (Intercept) 0.1943   0.4408
 Residual             1.5941   1.2626
```

```
Number of obs: 200, groups:  genotype, 20; loc, 10

Fixed effects:
                               Estimate Std. Error t value
(Intercept)                     -4.2757     0.6132  -6.972
poly(ctemperature, 2, raw = T)1  -2.1124     0.1659 -12.730
poly(ctemperature, 2, raw = T)2   0.5890     0.1889   3.119

Correlation of Fixed Effects:
            (Intr) p(,2,r=T)1
pl(,2,r=T)1  0.000
pl(,2,r=T)2 -0.306  0.000
>
```

The outcome of the mixed-effects model is the linear effect of temperature on flowering phenology, whilst allowing the intercepts of each genotype's relative date of flowering to account for some of the residual variance in the model.

```
> r.squaredGLMM(model1.3)
           R2m       R2c
[1,] 0.3699633 0.8753136
>
```

Again, we are interested primarily in the value of `R2c`, which is a substantially improved model fit over the simpler model1.2 (R^2 = 0.37 increased to R^2 = 0.88). Note that the `R2m` value of model1.3 is close to the R^2 of model1.2, but not identical because the inclusion of the random variable conditions the fixed-effects component of the model.

By applying the `re.form` function to model1.3, we can predict the average population-level reaction norm when `re.form = NA`, as well as the genotype reaction norms when `re.form = ~(1|genotype)`. Note that specifying `re.form = NULL` will fit all random effects, but with multiple random effects, the predictions can be nonsensical and most of the time visualising predictions from random effects independently will be preferable. We can then visualise both the overall population-level effect model prediction (as before) and the predicted genotype reaction norms from model1.3 as dashed lines of the same colour as the raw data (Fig. D).

```
> # Predict values based on the model fit using the predict function
> temperature_pred$fit1.3 <- predict(model1.3, newdata = temperature_pred, re.form =
NA)
> # Make a prediction for the population-level mean reaction norm
> # and append it to the flowerdata dataset
> flowerdata$pred_pop1.3  <- predict(model1.3, re.form = NA)
> # Make predictions for each genotype-level reaction norm
```

```
> flowerdata$pred_geno1.3 <- predict(model1.3, re.form = ~(1|genotype))
> # Plot predicted genotype reaction norms over the raw data, along with the overall
mean
> ggplot(temperature_pred, aes(x = ctemperature, y = fit1.3)) +
+   geom_line(data = flowerdata, aes(y = pred_geno1.3, group = genotype, colour =
genotype), lty = 2) +
+   geom_line(data = flowerdata,
+             aes(y = relativedate, group = genotype, colour = genotype)) +
+   geom_line(size = 2) +
+   ylab("Relative date of flowering") + xlab("Mean-centred growth temperature") +
+   theme_classic()
>
```

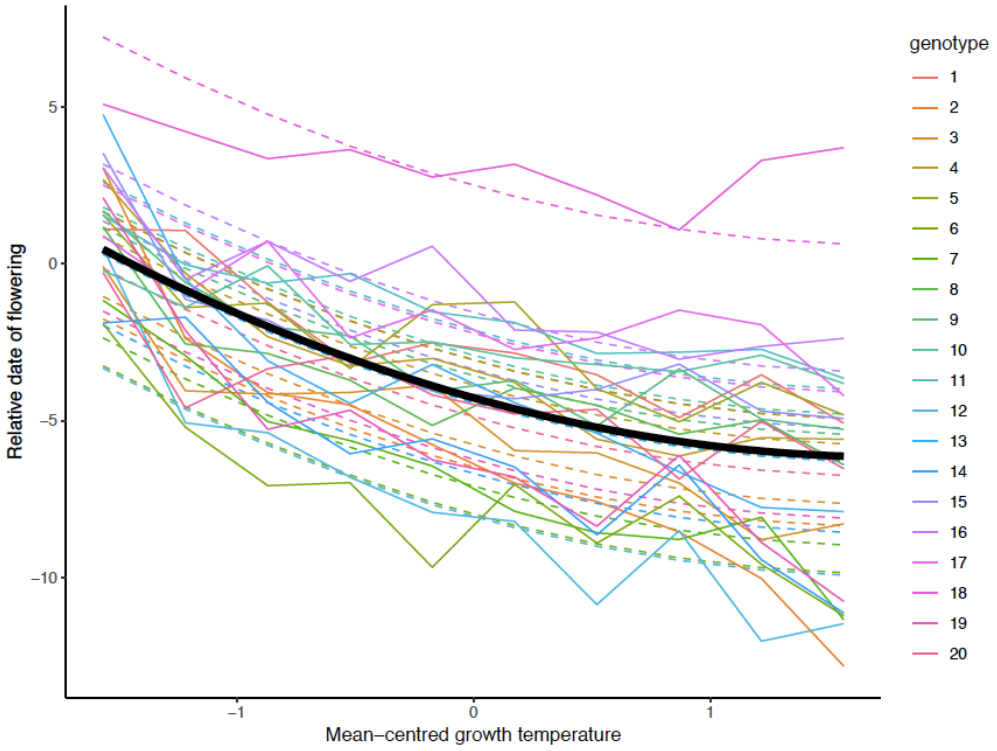

Figure D. Plot of relative date of flowering across mean-centred growth temperatures for 20 genotypes. Solid coloured lines represent the raw data, the thick black line represents the quadratic regression model fit (model1.3) of the overall effect of mean-centred growth temperature (the predicted average population-level reaction norm), and dashed coloured lines represent each genotype's modelled reaction norms from the random intercepts-only mixed-effects model.

We have evaluated that the additional random term that allows genotypes to vary by intercepts increases the overall model fit in terms of R^2 values, but it is worth comparing the AIC values of the models again to confirm that increasing model complexity is worthwhile for explaining the patterns observed in the data.

```
> # Does adding genotype as a random intercept improve model fit?
> # Likelihood ratio test
> chi2 <- 2*(summary(model1.3)$logLik - summary(model1.2)$logLik)
```

```
> 1-pchisq(chi2, 1)
'log Lik.' 0 (df=6)
> # AIC comparison
> AIC(model1.1, model1.2, model1.3)
         df      AIC
model1.1  4 998.2138
model1.2  5 993.9486
model1.3  6 757.9457
>
```

Adding the random intercepts term in model1.3 explains far more residual variance than model1.2 without trading-off against increased model complexity: so much so that the *p*-value for the LRT is essentially zero and the difference in AIC values between model1.2 and model1.3 is huge. However, the random intercepts-only model does not necessarily capture the shape of the specific genotype reaction norms that well. For example, the relative date of flowering is overestimated at lower temperatures and underestimated at higher temperatures by the random intercepts-only model fit predictions (Fig. E), which is particularly noticeable in genotype 18 (which has the highest y-values). To evaluate whether model fit can be improved further and specifically to see whether the mismatch in predicted and observed reaction norms can be addressed, we can allow the slopes of genotypes to vary in addition to the intercepts, so that the random regression slopes might be fit better to the observed patterns in the raw data. The addition of '`+x-variable`' (in this example: '`1+ctemperature`') to the left side of the random effect term ('`|genotype`') in model1.2 allows the slopes of the random genotype regressions to vary across mean-centred growth temperature. Thus, the random component of the random regression mixed model is now written as `(1+ctemperature|genotype)`, which specifies that the random component that genotype can vary both in intercept and in slope.

```
> #### Quadratic fixed effects with linear random regression model ####
> # Fit a linear mixed effects model for the fixed effect of growth temperature on
> # relative date of flowering and random effect of genotype intercepts and slopes
> model1.4 <- lmer(relativedate ~ poly(ctemperature, 2, raw = T) + (1|loc) +
(1+ctemperature|genotype),
+                  REML = FALSE, data = flowerdata)
> # Check model summary and R squared values
> summary(model1.4)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: relativedate ~ poly(ctemperature, 2, raw = T) + (1 | loc) + (1 +
ctemperature | genotype)
   Data: flowerdata

     AIC      BIC   logLik deviance df.resid
   685.2    711.5   -334.6    669.2      192
```

```
Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.39982 -0.57568 -0.02542  0.44598  2.64266

Random effects:
 Groups   Name         Variance Std.Dev. Corr
 genotype (Intercept)  6.3412   2.5182
          ctemperature 0.6190   0.7868   0.80
 loc      (Intercept)  0.2383   0.4882
 Residual              0.8806   0.9384
Number of obs: 200, groups:  genotype, 20; loc, 10

Fixed effects:
                                Estimate Std. Error t value
(Intercept)                      -4.2757     0.6178  -6.921
poly(ctemperature, 2, raw = T)1  -2.1124     0.2436  -8.673
poly(ctemperature, 2, raw = T)2   0.5890     0.1917   3.072

Correlation of Fixed Effects:
            (Intr) p(,2,r=T)1
pl(,2,r=T)1  0.529
pl(,2,r=T)2 -0.309  0.000
>
> r.squaredGLMM(model1.4)
          R2m       R2c
[1,] 0.369358 0.9312331
>
```

The value of `R2c` suggests that the random regression model1.4 has improved the fit over the random intercepts-only model1.3 from $R^2 = 0.88$ to $R^2 = 0.93$. Again, the predictions from model1.4 can then be plotted to visualise the random regressions that were modelled for each genotype as separate dashed lines of the same colour as the raw data, in addition to the overall population-level effect (Fig. E).

```
> # Predict values based on the model fit using the predict function
> temperature_pred$fit1.4 <- predict(model1.4, newdata = temperature_pred, re.form =
NA)
> # Make a prediction for the population-level mean reaction norm and append it to the
flowerdata dataset
> flowerdata$pred_pop1.4  <- predict(model1.4, re.form = NA)
> # Make predictions for the genotype-level reaction norms
> flowerdata$pred_geno1.4 <- predict(model1.4, re.form = ~(1+ctemperature|genotype))
> # Plot predicted genotype reaction norms over the raw data, along with the overall
mean
> ggplot(temperature_pred, aes(x = ctemperature, y = fit1.4)) +
```

```
+   geom_line(data = flowerdata, aes(y = pred_geno1.4, group = genotype, colour =
genotype), lty = 2) +
+   geom_line(data = flowerdata, aes(y = relativedate, group = genotype, colour =
genotype)) +
+   geom_line(size = 2) +
+   ylab("Relative date of flowering") + xlab("Mean-centred growth temperature") +
+   theme_classic()
>
```

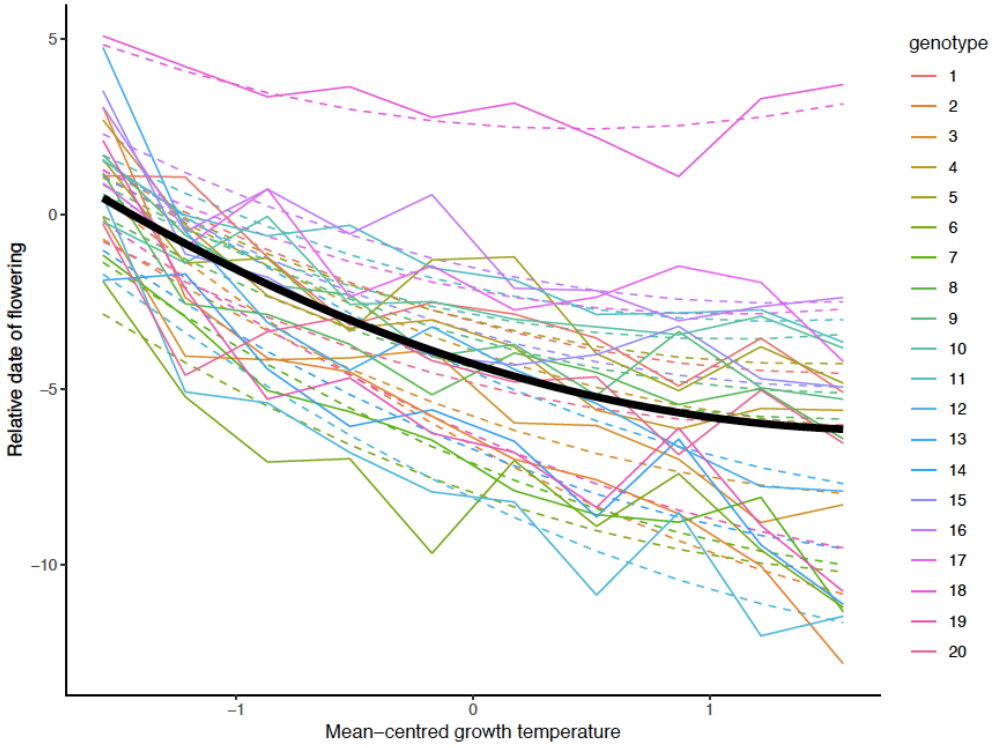

Figure E. Plot of relative date of flowering across mean-centred growth temperatures for 20 genotypes. Solid coloured lines represent the raw data, the thick black line represents the quadratic regression model fit (model1.4) of the overall effect of mean-centred growth temperature (the predicted average population-level reaction norm), and dashed coloured lines represent each genotype's modelled reaction norms from the random regression mixed-effects model.

In addition to the improved R^2, the reaction norms for each genotype predicted by the random regression model are also an improved visual fit to the raw data (Fig. E). If the LRT indicates that the models are significantly different and the AIC values continued to become lower, then the model fit is improved even when trading off increased model complexity, and we may want to try extending the model further.

```
> # Does adding genotype as a random intercept and slope further improve model fit?
> # Likelihood ratio test
> chi2 <- 2*(summary(model1.4)$logLik - summary(model1.3)$logLik)
> # The df difference between models can be checked by looking at the df within the
models being compared
> summary(model1.3)$logLik
```

```
'log Lik.' -372.9728 (df=6)
> summary(model1.4)$logLik
'log Lik.' -334.5753 (df=8)
> # Note that between model1.3 and model1.4 there is a change of 2 df, so the
> # pchisq change needs to be specified with 2 df rather than 1 as in previous
comparisons.
> 1-pchisq(chi2, 2)
'log Lik.' 0 (df=8)
> # AIC comparison
> AIC(model1.1, model1.2, model1.3, model1.4)
         df      AIC
model1.1  4 998.2138
model1.2  5 993.9486
model1.3  6 757.9457
model1.4  8 685.1506
>
```

Above, the df for each model was checked prior to calculating the *p*-value for the LRT, which is a difference of df = 2 in this comparison. Again, the LRT and AIC values suggests that the random regression mixed model has significantly improved the model fit to the data.

A final attempt to improve the model fit is to allow the random effect of genotype to vary in not only intercept and slope, but also in curvature, by fitting an additional quadratic random effect term.

```
> #### Quadratic fixed effects with quadratic random regression model ####
> # Fit a linear mixed effects model for the fixed effect of growth temperature on
> # relative date of flowering and random effect of genotype intercepts, slopes, and
curvature
> model1.5 <- lmer(relativedate ~ poly(ctemperature, 2, raw = T) + (1|loc) +
+                    (1 + ctemperature + I(ctemperature^2)|genotype),
+                  REML = FALSE, data = flowerdata)
> # Check model summary and R squared values
> summary(model1.5)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: relativedate ~ poly(ctemperature, 2, raw = T) + (1 | loc) + (1 +
    ctemperature + I(ctemperature^2) | genotype)
   Data: flowerdata

     AIC      BIC   logLik deviance df.resid
   687.7    724.0   -332.8    665.7      189

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.64712 -0.57304 -0.00154  0.49049  2.18010

Random effects:
 Groups   Name              Variance Std.Dev. Corr
```

```
 genotype (Intercept)       6.47564  2.5447
          ctemperature      0.62308  0.7894    0.84
          I(ctemperature^2) 0.02596  0.1611   -0.19 -0.70
 loc      (Intercept)       0.24037  0.4903
 Residual                   0.85311  0.9236
Number of obs: 200, groups:  genotype, 20; loc, 10

Fixed effects:
                                Estimate Std. Error t value
(Intercept)                      -4.2757     0.6234  -6.859
poly(ctemperature, 2, raw = T)1  -2.1124     0.2441  -8.653
poly(ctemperature, 2, raw = T)2   0.5890     0.1953   3.016

Correlation of Fixed Effects:
            (Intr) p(,2,r=T)1
pl(,2,r=T)1  0.552
pl(,2,r=T)2 -0.334 -0.093
>
> r.squaredGLMM(model1.5)
           R2m       R2c
[1,] 0.3693149 0.9333907
>
```

The value of `R2c` suggests that the random regression that also allows genotype to vary in quadratic curvature has very marginally improved the model fit over the standard random regression model1.4 from $R^2 = 0.931$ to $R^2 = 0.933$. Again, the predictions from model1.5 can then be plotted to visualise the modelled random regressions for each genotype as separate dashed lines of the same colour as the raw data, in addition to the population-level average reaction norm (Fig. F). To visualise the model predictions for such a complex random effect structure, an alternative model without the additional random effect of (1|loc) is specified so that the predict function works correctly. We can check whether omitting the (1|loc) random effect changes the fixed effect coefficients greatly before interpreting the plot without the term.

```
> # Predict values based on the model fit using the predict function
> temperature_pred$fit1.5 <- predict(model1.5, newdata = temperature_pred, re.form =
NA)
> # Make a prediction for the population-level mean reaction norm and append it to the
flowerdata dataset
> flowerdata$pred_pop1.5  <- predict(model1.5, re.form = NA)
> # Unfortunately, to coerce the predict function to work for a complex random effect,
> # the model needs to be specified without the second random effect (1|loc)
> model1.5a <- lmer(relativedate ~ poly(ctemperature, 2, raw = T) +
+                     (1 + ctemperature + I(ctemperature^2)|genotype),
+                   REML = FALSE, data = flowerdata)
```

```
> # We can check whether omitting the (1|loc) random effect changes the fixed effect
> # coefficients greatly before interpreting the plot without it
> summary(model1.5)$coef
                                Estimate Std. Error   t value
(Intercept)                   -4.2757440  0.6233534 -6.859262
poly(ctemperature, 2, raw = T)1 -2.1124107  0.2441291 -8.652844
poly(ctemperature, 2, raw = T)2  0.5889973  0.1953206  3.015542
> summary(model1.5a)$coef
                                Estimate Std. Error    t value
(Intercept)                   -4.2757440 0.57813252  -7.395785
poly(ctemperature, 2, raw = T)1 -2.1124107 0.18726887 -11.280095
poly(ctemperature, 2, raw = T)2  0.5889973 0.09162355   6.428449
>
> # Make predictions for the genotype-level reaction norms
> flowerdata$pred_geno1.5 <- predict(model1.5a, re.form = NULL)
> ggplot(temperature_pred, aes(x = ctemperature, y = fit1.5)) +
+   geom_line(data = flowerdata, aes(y = pred_geno1.5, group = genotype, colour =
genotype), lty = 2) +
+   geom_line(data = flowerdata, aes(y = relativedate, group = genotype, colour =
genotype)) +
+   geom_line(size = 2) +
+   ylab("Relative date of flowering") + xlab("Mean-centred growth temperature") +
+   theme_classic()
>
```

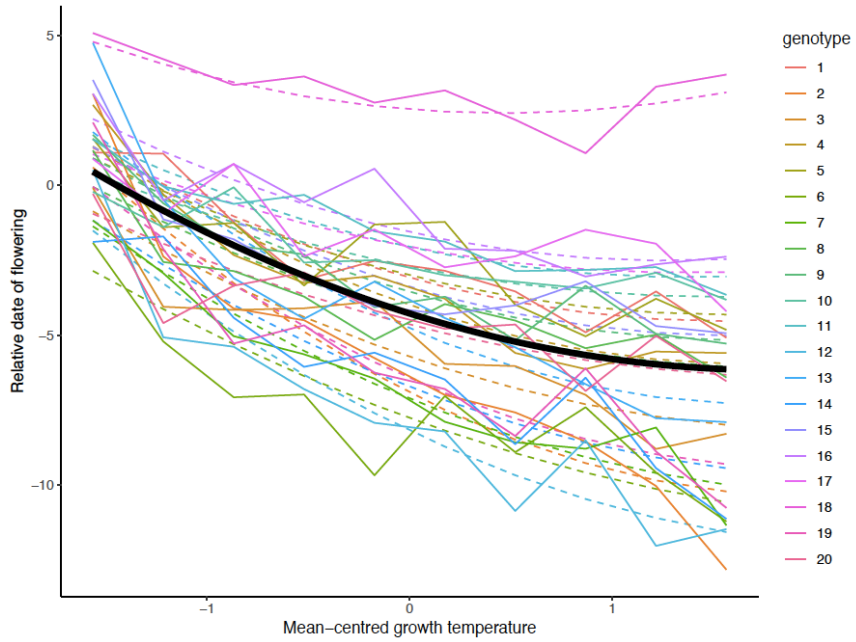

Figure F. Plot of relative date of flowering across mean-centred growth temperatures for 20 genotypes. Solid coloured lines represent the raw data, the thick black line represents the quadratic regression model fit (model1.5) of the overall effect of mean-centred growth temperature (the predicted average population-level reaction norm), and dashed coloured lines represent each genotype's modelled reaction norms from the random regression mixed-effects model that accounts for differences in intercept, slope, and quadratic curvature.

The LRT can be applied to compare model1.4 and model1.5 using df = 3, and the AIC values for the five models can now be compared as well.

```
> # Does adding genotype as a random intercept, slope, and curvature further improve
model fit?
> # Likelihood ratio test
> chi2 <- 2*(summary(model1.5)$logLik - summary(model1.4)$logLik)
> # The df difference between models can be checked by looking at the df within the
models being compared
> summary(model1.4)$logLik
'log Lik.' -334.5753 (df=8)
> summary(model1.5)$logLik
'log Lik.' -332.8359 (df=11)
> # Note that between model1.3 and model1.4 there is a change of 3 df, so the
> # pchisq change needs to be specified with 3 df rather than 1 or 2 as in previous
comparisons.
> 1-pchisq(chi2, 3)
'log Lik.' 0.3235111 (df=11)
> # AIC comparison
> AIC(model1.1, model1.2, model1.3, model1.4, model1.5)
         df      AIC
model1.1  4 998.2138
model1.2  5 993.9486
model1.3  6 757.9457
model1.4  8 685.1506
model1.5 11 687.6717
>
```

The additional complexity of curvature in model1.5 results in a non-significant LRT *p*-value and a larger AIC value over model1.4, so the slight improvement of model fit in terms of R^2 comes at a cost of model parsimony. At this point, we have compared five models of increasing complexity and quantitatively determined that of these models, model1.4 is the best model for these data. We will proceed hereafter with model1.4 to demonstrate how a random regression model can be used to rank plasticity, if desired, by extracting the best linear unbiased predictors (BLUPs) for each genotype.

Using BLUPs to rank plasticity

Before demonstrating how BLUPs could be used to rank plasticity, we must begin with a formal warning. BLUPs estimated from linear mixed-effects regression models fitted in `lme4` are single point estimates that do not have associated measures of uncertainty. As a result, any derived statistics or formal interpretation of plasticity based on these BLUPs is potentially very dangerous and anti-conservative without properly accounting for estimation uncertainty. For using BLUPs beyond simple ranking (e.g., of the least to most plastic genotypes), we strongly encourage readers to read the references we provide here to avoid the misuse of BLUPs by using a Bayesian MCMC framework (e.g., by using the `MCMCglmm` package in `R`) to generate estimates of uncertainty around BLUPs (Hadfield, 2010; Hadfield *et al.*, 2010; Houslay & Wilson, 2017).

With the cautionary points outlined above in mind, we can proceed to extract BLUPs to rank which genotypes are least or most plastic. BLUPs can be extracted from a mixed-effects models by calling `ranef()`. Here, BLUPs represent the response of a given genotype to the fixed effect of temperature as the difference between that genotype's predicted response and the population-level average predicted response. In the case of a random regression model (as in model1.4), there are two random effects: the intercept and the slope. The BLUP intercept term indicates the difference in genotype elevation relative to the population-average, so more positive values of BLUP intercept indicate that the genotype's reaction norm occurs above the population-level average and negative values are below the population-level average (Fig. G).

```
> #### Best Linear Unbiased Predictors (BLUPs) to rank plasticity ####
> # BLUPs represent the response of a given genotype to the fixed effect of temperature
> # as the difference between that genotype's predicted response and the population-
level. Here, we calculate and plot BLUPs for ranking plasticity.
> # average predicted response. > genotype_blups <- ranef(model1.4)$`genotype`
> genotype_index <- as.factor(c(1:20))
> genotype_data  <- cbind(genotype_index, genotype_blups)
> colnames(genotype_data) <- c("genotype", "BLUP_int", "BLUP_slope")
>
> # BLUPs by intercept
> ggplot(genotype_data, aes(genotype, BLUP_int)) +
+   geom_point(aes(group = genotype, colour = genotype), size = 4) +
+   ylab("BLUP intercept estimate") +
+   geom_hline(yintercept = 0, lty = 2) + theme_classic()
>
```

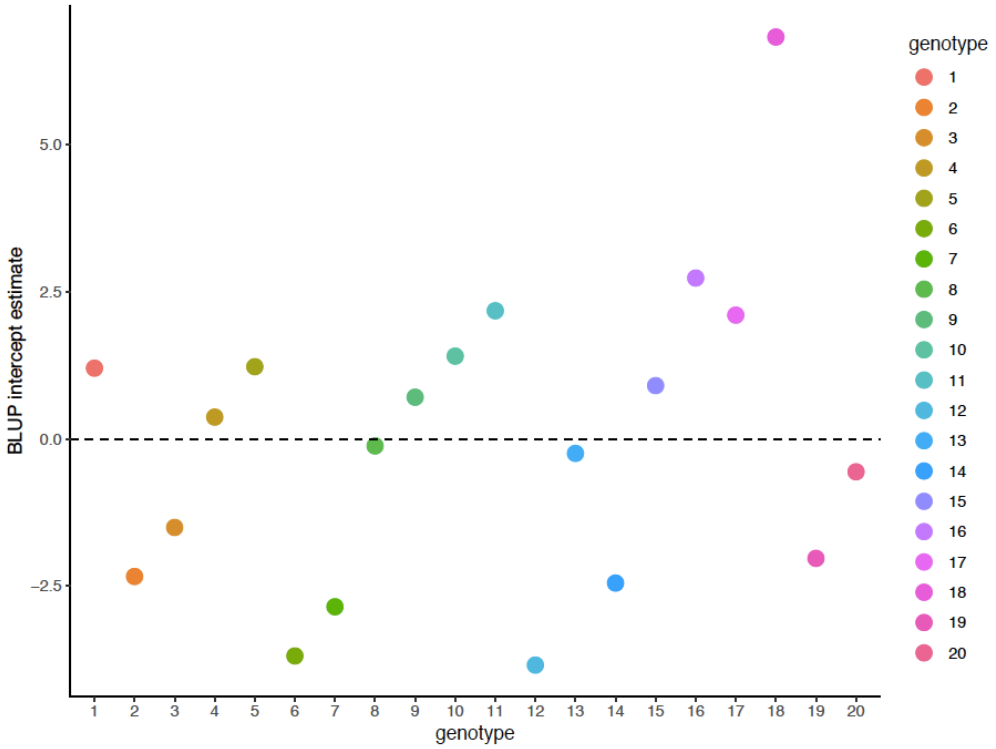

Figure G. BLUP intercept estimates for the relative date of flowering response to mean-centred growth temperatures of 20 genotypes. Values are relative to the overall population-level average response.

The BLUP intercept values are not a measure of plasticity, but these values may be correlated with BLUP slope values and otherwise may be a parameter of interest for comparing among genotypes. The BLUP slope estimate is the difference in slope (relative steepness of change) between the population-level average response and the response of the individual genotype. Here, that is the difference in slope of the relative date of flowering for each value of temperature *relative* to the population-level average slope (Fig. H).

```
> # BLUPs by slope
> ggplot(genotype_data, aes(genotype, BLUP_slope)) +
+   geom_point(aes(group = genotype, colour = genotype), size = 4) +
+   ylab("Plasticity (BLUP slope estimate)") +
+   geom_hline(yintercept = 0, lty = 2) + theme_classic()
>
```

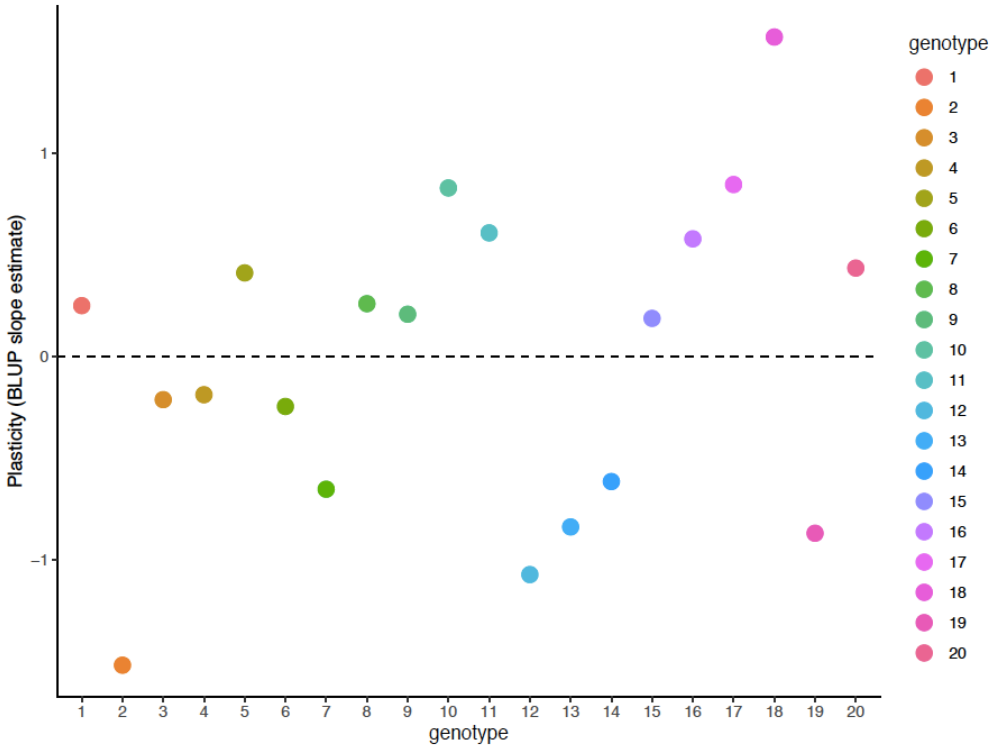

Figure H. BLUP slope estimates for the relative day of flowering response to mean-centred growth temperature of 20 genotypes. Values are relative to the overall population-level average response.

Because the population-level average response is negative overall, all genotypes have a negative slope when the BLUP slope estimates are added to the population-level average slope estimate from model1.4 (Fig. I). Note however, that while this approach is useful to visualise how BLUPs are relative to the population-level average, they do not account for the non-linear (quadratic fixed-effect) term in model1.4.

```
> # Add the BLUP slopes for the genotypes to the population average
> pop_av_slope <- summary(model1.4)$coefficients[2]
> genotype_data$genotype_slopes <- genotype_blups$ctemperature + pop_av_slope
> # BLUPs by slope + population-level average
> ggplot(genotype_data, aes(genotype, genotype_slopes)) +
+   geom_point(aes(group = genotype, colour = genotype), size = 4) +
+   ylab("Plasticity (population-average + BLUP slope estimate)") +
+   geom_hline(yintercept = 0, lty = 2) + theme_classic()
>
```

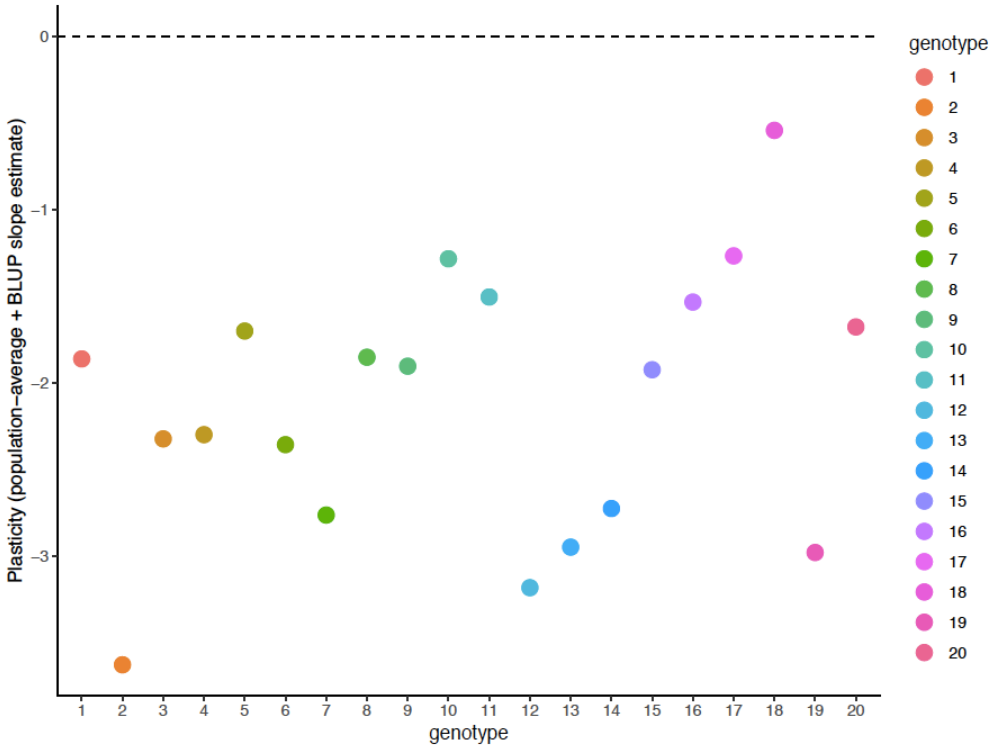

Figure I. Plasticity as the sum of the population-level average linear slope prediction added to the BLUP slope estimates for the relative day of flowering response to mean-centred growth temperature of 20 genotypes. All values are negative because the relative day of flowering becomes more negative across mean-centred growth temperatures for all 20 genotypes.

The BLUP intercept and slope estimates are sometimes correlated. The correlation coefficient is given in the random effects correlation from the model1.4 summary, which is 0.8. This positive relationship can clearly be seen in Fig. J, where the genotype with the most positive BLUP slope estimate has the highest positive intercept (Fig. J) and has the least plasticity across growth temperatures (Fig. E): once the population-level average linear slope is added to the BLUP slope estimate, this estimate is closest to zero among all 20 genotypes (Fig. I).

```
> # Correlation between BLUP intercepts and slopes
> ggplot(genotype_data, aes(BLUP_int, BLUP_slope)) +
+   geom_point(aes(group = genotype, colour = genotype), size = 4) +
+   xlab("BLUP intercept estimate") +
+   ylab("BLUP slope estimate") +
+   theme_classic()
>
```

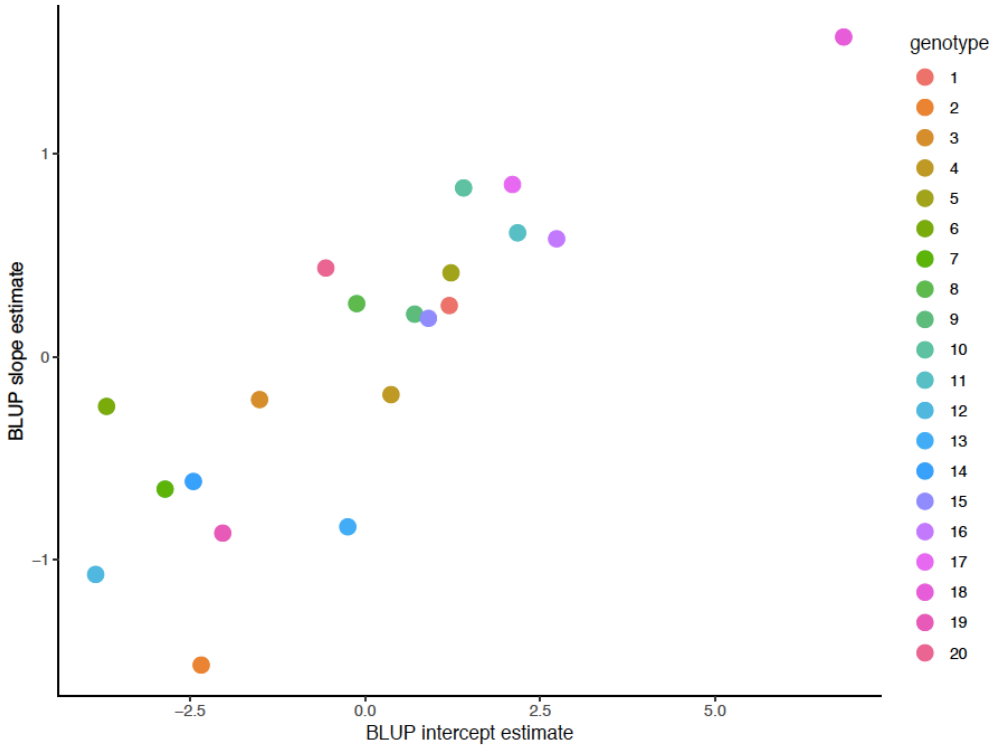

Figure J. Correlation between model1.4 BLUP intercept and slope estimates.

The genotypes can then be ranked in order of plasticity by BLUP slope estimates (Fig. K). As a reminder, because the population-level average response is negative, the most negative BLUP slope estimates represent steeper reaction norm slopes and hence greater plasticity, and more positive BLUP slope estimates represent flatter reaction norms and less plasticity in the relative date of flowering in response to growth temperatures.

```
> # Rank the BLUPs in order
> # Sort BLUPs by slope of most to least plastic
> genotype_data$genotype_ordered <- factor(genotype_data$genotype, levels =
genotype_data$genotype[order(genotype_data$BLUP_slope)])
> ggplot(genotype_data, aes(genotype_ordered, BLUP_slope)) +
+   geom_bar(stat = "identity", aes(group = genotype, fill = genotype)) +
+   xlab("Genotype (in ranked order of plasticity)") +
+   ylab("Plasticity (BLUP slope estimate)") +
+   theme_classic()
>
```

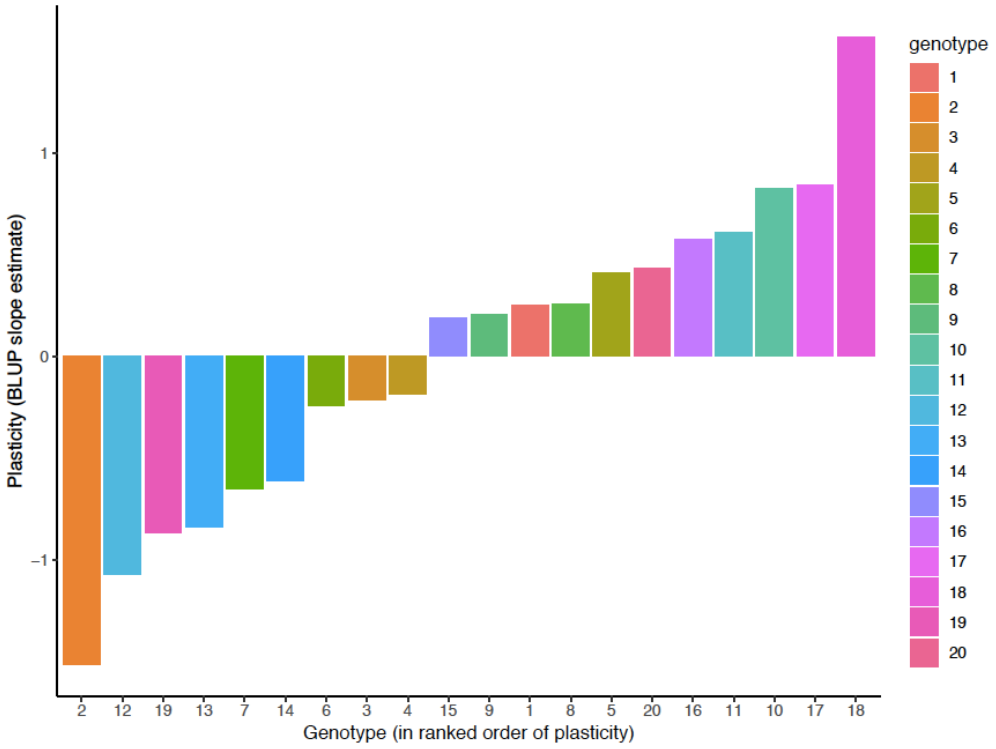

Figure K. Ranking genotypes in order of plasticity estimated as BLUP slope estimates. More negative BLUP slope estimates represent greater plasticity and more positive BLUP slope estimates represent less plasticity in the relative date of flowering in response to mean-centred growth temperatures.

Although the outcome is the same as ranking by BLUP slope estimates, it may be easier to make sense of plasticity ranking by visualising the sum of the BLUP slope estimates and the population-level average linear estimates (Fig. L).

```
> # Another way to visualise the plasticity rank for negative data is by adding
> # the BLUP slope values to the population-level average effect of temperature
> ggplot(genotype_data, aes(genotype_ordered, genotype_slopes)) +
+   geom_bar(stat = "identity", aes(group = genotype, fill = genotype)) +
+   xlab("Genotype (in ranked order of plasticity)") +
+   ylab("Plasticity (population-average + BLUP slope estimate)") +
+   theme_classic()
>
```

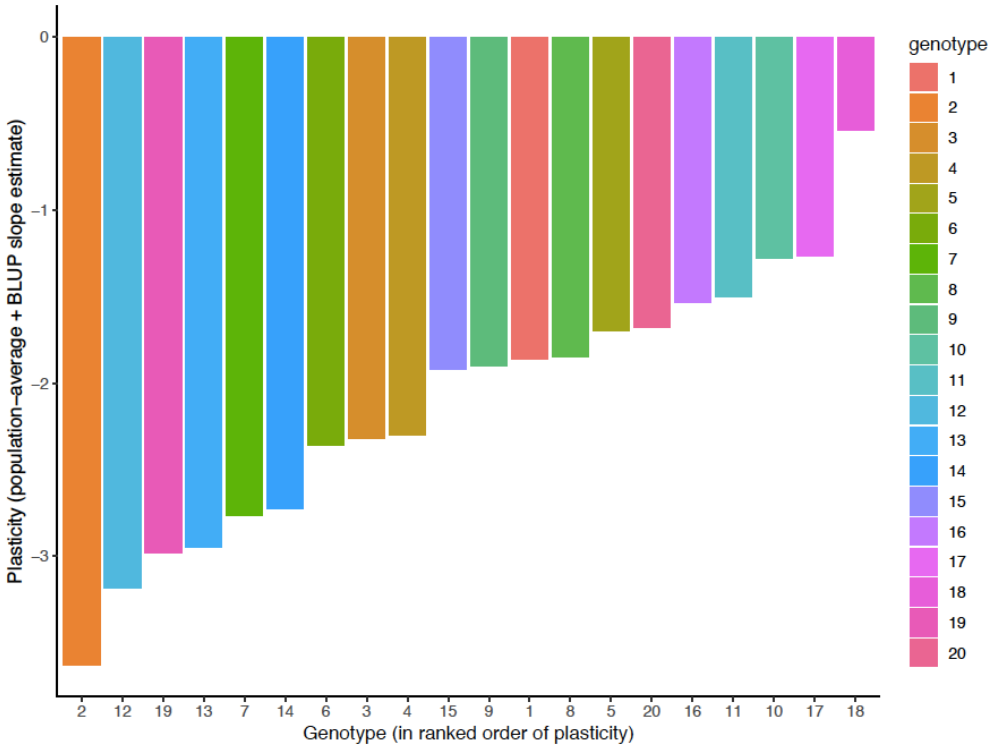

Figure L. Ranking genotypes in order of plasticity estimated the sum of the population-level average linear slope prediction added to the BLUP slope estimates. More negative BLUP slope estimates represent greater plasticity and more positive BLUP slope estimates represent less plasticity in relative date of flowering in response to mean-centred growth temperatures.

Summary

This tutorial has demonstrated the basic principles of building models up from simple linear models to random regression mixed models (RRMMs) with non-linear functions as both fixed and random effects for the analysis of phenotypic plasticity data. We have also shown how to visualise these models, how to use R^2 and AIC values to determine which model to proceed with, and how BLUPs can be used to rank genotypes by plasticity.

Supporting Information Notes S3

Dataset of flowering phenology plasticity used in the worked example tutorial ('flowerdata.csv').

Supporting Information Notes S4

R code for the worked example tutorial. Details for downloading and installing packages are provided at the beginning of Supporting Information Notes S2.

Required software: R and packages: ggplot2, MuMIn, and lme4.

Recommended software: RStudio

References

Bartoń KA 2018. Package 'MuMIn'. Multi-model inference. R package verision 1.42.1: http://cran.r-project.org/web/package=MuMIn).

Bates D, Mächler M, Bolker B, Walker S. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* **67**: 1-48.

Butler D, Cullis BR, Gilmour A, Gogel B. 2009. ASReml-R reference manual. *The State of Queensland, Department of Primary Industries and Fisheries, Brisbane*.

Crawley MJ. 2012. *The R book*. Oxford, UK: Wiley.

Hadfield JD. 2010. MCMC methods for multi-response generalized linear mixed models: the MCMCglmm R package. *Journal of Statistical Software* **33**: 1-22.

Hadfield JD, Wilson AJ, Garant D, Sheldon BC, Kruuk LEB. 2010. The misuse of BLUP in ecology and evolution. *The American Naturalist* **175**: 116-125.

Houslay TM, Wilson AJ. 2017. Avoiding the misuse of BLUP in behavioural ecology. *Behavioral Ecology* **28**: 948-952.

Jochner S, Sparks TH, Laube J, Menzel A. 2016. Can we detect a nonlinear response to temperature in European plant phenology? *International Journal of Biometeorology* **60**: 1551-1561.

Johnson PCD. 2014. Extension of Nakagawa & Schielzeth's R^2_{GLMM} to random slopes models. *Methods in Ecology and Evolution* **5**: 944-946.

Kirkpatrick M, Lofsvold D, Bulmer M. 1990. Analysis of the inheritance, selection and evolution of growth trajectories. *Genetics* **124**: 979-993.

Marchal A, Schlichting C, Gobin R, Balandier P, Millier F, Muñoz F, Paques L, Sanchez L. 2018. Deciphering hybrid larch reaction norms using random regression. *G3: Genes|Genomes|Genetics*. doi: 10.1534/g3.118.200697.

Martin JGA, Nussey DH, Wilson AJ, Réale D. 2011. Measuring individual differences in reaction norms in field and experimental studies: a power analysis of random regression models. *Methods in Ecology and Evolution* **2**: 362-374.

Morrissey MB, Liefting M. 2016. Variation in reaction norms: statistical considerations and biological interpretation. *Evolution* **70**: 1944-1959.

Nakagawa S, Schielzeth H. 2013. A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution* **4**: 133-142.

Paine CET, Marthews TR, Vogt DR, Purves D, Rees M, Hector A, Turnbull LA. 2011. How to fit nonlinear plant growth models and calculate growth rates: an update for ecologists. *Methods in Ecology and Evolution* **3**: 245-256.

Petavy G, David JR, Gibert P, Moreteau B. 2001. Viability and rate of development at different temperatures in *Drosophila*: a comparison of constant and alternating thermal regimes. *Journal of Thermal Biology* **26**: 29-39.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Development Core Team. 2014. nlme: linear and nonlinear mixed effects models. *R package version 3.1-117*.

R Development Core Team. 2017. R: a language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria.

Richards FJ. 1959. A flexible growth function for empirical use. *Journal of Experimental Botany* **10**: 290-301.

Stratton DA. 1998. Reaction norm functions and QTL–environment interactions for flowering time in *Arabidopsis thaliana*. *Heredity* **81**: 144-155.

van de Pol M. 2012. Quantifying individual variation in reaction norms: how study design affects the accuracy, precision and power of random regression models. *Methods in Ecology and Evolution* **3**: 268-280.

van de Pol M, Wright J. 2009. A simple method for distinguishing within- versus between-subject effects using mixed models. *Animal Behaviour* **77**: 753-758.

Wood SN. 2017. *Generalized additive models: an introduction with R*. Boca Raton, FL, USA: Chapman and Hall/CRC Press, Taylor & Francis Group.

Zuur AF, Ieno EN, Walker NJ, Saveliev AA, Smith GM. 2009. *Mixed effects models and extensions in ecology with R*. New York, NY, USA: Springer.