

Calibration for *Araucaria araucana*

This is an example of the simple NLME model of *Araucaria araucana* being calibrated in a new stand first and then tree height predicted using the calibrated NLME model for the new stand. In this example, the new stand is stand 2 in the validating database.

Packages involved in the calibration

```
1 library(nlme)
2 library(dplyr)
```

Calibration and height prediction

- Simple NLME model developed (SMAA)

```
1 Fitting_AA <- read.csv("Anpassung_AA.csv")
2 colnames(Fitting_AA) <- c('Nr.', 'Site', 'Stand', 'Plot', 'Species', 'd',
3   'h')
4 SMAA <- nlme(h ~ 1.3 + a * exp(-b * exp(-c * d)), start = c(a =
5   46.410000000, b = 2.174185939, c = 0.008847333), data = Fitting_AA,
6   fixed = a + b + c ~ 1, random = a ~ 1 | Stand, weights = varPower(form
7   = ~ d))
8
9 print(SMAA)
```

```
1 ## Nonlinear mixed-effects model fit by maximum likelihood
2 ##   Model: h ~ 1.3 + a * exp(-b * exp(-c * d))
3 ##   Data: Fitting_AA
4 ##   Log-likelihood: -1105.2
5 ##   Fixed: a + b + c ~ 1
6 ##           a           b           c
7 ## 25.80305215  2.64794801  0.02519394
8 ##
9 ## Random effects:
10 ##   Formula: a ~ 1 | Stand
11 ##           a   Residual
12 ## StdDev: 1.421823 0.8851582
13 ##
14 ## Variance function:
15 ##   Structure: Power of variance covariate
16 ##   Formula: ~d
17 ##   Parameter estimates:
18 ##       power
19 ## 0.3540389
```

```

20 ## Number of Observations: 392
21 ## Number of Groups: 10

```

- Calling the data in SMAA

All results are printed so that you can use them in your stand or plot.

```

1 .G <- VarCorr(SMAA, rdig = 7) # read variance and correlation components
2 print(.G)

```

```

1 ## Stand = pdLogChol(list(a ~ 1))
2 ##           Variance StdDev
3 ## a           2.021580 1.4218229
4 ## Residual    0.783505 0.8851582

```

```

1 .estimate <- as.numeric(.G[, 1])
2 print(.estimate)

```

```

1 ## [1] 2.021580 0.783505

```

```

1 .cov_parm <- c('var_u', 'Residual')
2 print(.cov_parm)

```

```

1 ## [1] "var_u"      "Residual"

```

```

1 .CEP <- data.frame(.cov_parm, .estimate)
2 print(.CEP)

```

```

1 ##   .cov_parm .estimate
2 ## 1      var_u 2.021580
3 ## 2  Residual 0.783505

```

```

1 var_e <- .CEP[2, 2] # sigma^2: the scaling factor, given by the value of
2 print(var_e)

```

```

1 ## [1] 0.783505

```

```

1 D <- .CEP[1, 2] # D: the structure of the variance-covariance matrix among
2 print(D)

```

```

1 ## [1] 2.02158

```

```

1 parms <- data.frame(t(SMAA$coefficients$fixed)) # estimated values for the
2 print(parms)

```

```

1 ##           a           b           c
2 ## 1 25.80305 2.647948 0.02519394

```

```

1 rho <- coef(SMAA$modelStruct$varStruct, un = FALSE) # the coefficient of
  the variance function
2 print(rho)

```

```

1 ##      power
2 ## 0.3540389

```

- Calibration

Stand 2 in the validating database is used as the new stand in this example. For the simple NLME model, the calibration design is a random selection of 5 trees, so now we create two datasets, one with 5 sample trees in stand 2 and the other with those trees in stand 2 other than the five sample trees.

```

1 sample_d <- c(58, 139, 73, 207, 166)
2 sample_h <- c(18, 25, 21, 34, 36)
3 sample <- data.frame(sample_d, sample_h)
4
5 other_d <- c(57, 139, 139, 33, 61, 81, 166, 52, 207, 207, 166, 207, 166,
  98.7, 33, 207, 89)
6 other_h <- c(13, 25, 25, 13, 16, 30, 36, 13, 44, 44, 36, 34, 36, 22, 28,
  34, 20) # for your new stand, heights are not needed here, I am only
  showing them here to show you the predictive performance of the
  calibrated model in this example
7 other <- data.frame(other_d, other_h)

```

```

1 matcor <- diag((sample$sample_d) ^ (2 * rho), ncol = 5, nrow = 5) # Gi:
  diagonal matrix describing the nonconstant variance
2 r = var_e * matcor # Ri: within-stand variance-covariance matrix
3 y <- as.matrix(sample$sample_h)
4 Z <- as.matrix(exp(-parms$b * exp(-parms$c * sample$sample_d))) # the
  partial derivatives with respect to random parameters
5 fxBb <- as.matrix(1.3 + (parms$a) * exp(-parms$b * exp(-parms$c * sample$
  sample_d))) # the estimated tree height only with fixed effects
6 bi <- D %*% t(Z) %*% solve(r + Z %*% D %*% t(Z)) %*% ((y - fxBb)) # the
  estimated value of the random parameter --> Eq.24 in text
7 print(bi)

```

```

1 ##      [,1]
2 ## [1,] 1.178264

```

- height prediction

```

1 other$pre.h <- 1.3 + parms$a * exp(-parms$b * exp(-parms$c * other$other_d
  )) + bi * exp(-parms$b * exp(-parms$c * other$other_d)) # Eq.25 in text
2 print(other)

```

```

1 ##      other_d other_h      pre.h
2 ## 1      57.0      13 15.672149
3 ## 2     139.0      25 26.211759
4 ## 3     139.0      25 26.211759
5 ## 4      33.0      13  9.817512
6 ## 5      61.0      16 16.566678
7 ## 6      81.0      30 20.426590

```

8	## 7	166.0	36	27.212456
9	## 8	52.0	13	14.506853
10	## 9	207.0	44	27.895888
11	## 10	207.0	44	27.895888
12	## 11	166.0	36	27.212456
13	## 12	207.0	34	27.895888
14	## 13	166.0	36	27.212456
15	## 14	98.7	22	22.946958
16	## 15	33.0	28	9.817512
17	## 16	207.0	34	27.895888
18	## 17	89.0	20	21.666363