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 12 in the validating database.

Packages involved in the calibration

```
library(nlme)
library(dplyr)
```

Calibration and height prediction

- Simple NLME model developed (SMAA)

```
Fitting_AA <- read.csv("/Users/siiiina/Desktop/code/Anpassung_AA.csv")
colnames(Fitting_AA) <- c('Nr.', 'Site', 'Stand', 'Plot', 'Species', 'd',
    'h')

SMAA <- nlme(h ~ 1.3 + a * exp(-b * exp(-c * d)),start = c(a =
    46.410000000, b = 2.174185939, c = 0.008847333), data = Fitting_AA,
    fixed = a + b + c ~ 1, random = a ~ 1 | Stand, weights = varPower(form
    = ~ d))

print(SMAA)
```

```
## Nonlinear mixed-effects model fit by maximum likelihood
##   Model: h ~ 1.3 + a * exp(-b * exp(-c * d))
##   Data: Fitting_AA
##   Log-likelihood: -1105.2
##   Fixed: a + b + c ~ 1
##           a           b           c
## 25.80305209  2.64794802  0.02519394
##
## Random effects:
##  Formula: a ~ 1 | Stand
##                a  Residual
## StdDev: 1.421823 0.8851574
##
## Variance function:
##  Structure: Power of variance covariate
##  Formula: ~d
##  Parameter estimates:
##     power
## 0.3540391
```

```
## Number of Observations: 392
## Number of Groups: 10
```

- Calling the data in SMAA

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

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

```
## Stand = pdLogChol(list(a ~ 1))
##          Variance  StdDev
## a        2.0215805 1.4218229
## Residual 0.7835037 0.8851574
```

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

```
## [1] 2.0215805 0.7835037
```

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

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

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

```
##   .cov_parm .estimate
## 1     var_u 2.0215805
## 2  Residual 0.7835037
```

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

```
## [1] 0.7835037
```

```
D <- .CEP[1, 2] # D: the structure of the variance-covariance matrix among
    stands, since there is only one random parameter in SMAA, D is the
   variance of this parameter
print(D)
```

```
## [1] 2.02158
```

```
parms <- data.frame(t(SMAA$coefficients$fixed)) # estimated values for the
    fixed parameters of SMAA
print(parms)
```

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

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

```
##     power
## 0.3540391
```

- Calibration

Stand 12 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 12 and the other with those trees in stand 12 other than the five sample trees.

```r
sample_d <- c(172, 42, 18, 82, 114)
sample_h <- c(34.5, 11, 8.5, 20, 25)
sample <- data.frame(sample_d, sample_h)

other_d <- c(73, 53, 74, 74, 85, 64, 27, 92, 93, 86, 82, 40, 82, 28, 78, 83, 94, 124, 73.5, 78, 47, 91, 74, 9, 30, 77.4, 94, 50, 80.5, 238, 65, 91)
other_h <- c(18, 15, 20, 19, 19, 5, 9, 19, 24, 16, 18, 12, 21, 6, 18, 17, 14, 29, 20.5, 27, 10, 27, 19, 4.5, 6.5, 21.5, 15, 17, 26, 35, 18, 20) # for your new stand, heights are not measured here, I am only showing them here to show you the predictive performance of the calibrated model in this example
other <- data.frame(other_d, other_h)
```

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

```
##           [,1]
## [1,] 0.6353685
```

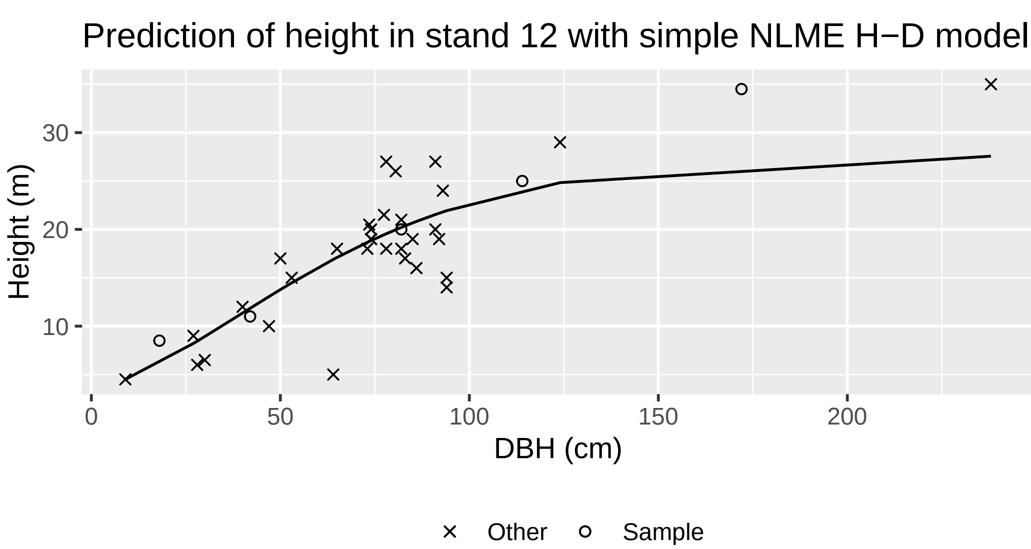
- height prediction

```r
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
print(other)
```

```
##    other_d other_h     pre.h
## 1     73.0    18.0 18.655747
## 2     53.0    15.0 14.473187
## 3     74.0    20.0 18.838443
## 4     74.0    19.0 18.838443
```

```
## 5      85.0     19.0 20.670218
## 6      64.0      5.0 16.892680
## 7      27.0      9.0  8.214634
## 8      92.0     19.0 21.669384
## 9      93.0     24.0 21.801973
## 10     86.0     16.0 20.820715
## 11     82.0     18.0 20.202860
## 12     40.0     12.0 11.356512
## 13     82.0     21.0 20.202860
## 14     28.0      6.0  8.449250
## 15     78.0     18.0 19.542246
## 16     83.0     17.0 20.361305
## 17     94.0     14.0 21.932094
## 18    124.0     29.0 24.832086
## 19     73.5     20.5 18.747431
## 20     78.0     27.0 19.542246
## 21     47.0     10.0 13.057622
## 22     91.0     27.0 21.534302
## 23     74.0     19.0 18.838443
## 24      9.0      4.5  4.502963
## 25     30.0      6.5  8.923782
## 26     77.4     21.5 19.439432
## 27     94.0     15.0 21.932094
## 28     50.0     17.0 13.772044
## 29     80.5     26.0 19.960175
## 30    238.0     35.0 27.564794
## 31     65.0     18.0 17.098867
## 32     91.0     20.0 21.534302
```

- visualisation



As mentioned in the paper, the model developed on the basis of Gompertz model appears to underestimate the height of larger diameter trees more or less when predicting tree height. Please note this in practice.