

Biphasic von Bertalanffy growth curve for common sole using non-linear mixed effects models (SAEM Algorithm)

The existence of a trade-off between allocating energy between somatic growth and reproduction has been suggested (Lester et al., 2004). Reproductive effort should negatively influence growth: more energy would be allocated to somatic growth during the young years of life (i.e. immature fish), leading to fast growth, whereas after reaching sexual maturity, energy would be divided into two activities (reproductive investment and somatic growth), and the growth in size would decrease as a consequence. Hence, biphasic growth curves to correct for the absence of energetic costs linked to reproduction before sexual maturation (or the small energetic cost during the first few years after maturation) have been proposed (Day and Taylor, 1997; Lester et al., 2004; Charnov, 2008; Quince et al., 2008a, b).

Until now (using standard non linear models as `nls` in `stat` package), the individual parameters were considered a fixed effects: we didn't make any assumption about their possible values. In a population approach, the N subjects are assumed to be randomly sampled from a same population of individuals. Then, each individual parameter is treated as a random variable. A population approach and the use of mixed effects models will allow us to take into account this inter individual variability.

Several algorithms exist for maximum likelihood estimation in nonlinear mixed effects models. In particular, the stochastic approximation EM algorithm (SAEM) is an iterative algorithm that converges to a maximum of the likelihood function under general conditions. The `saemix` package for R provides maximum likelihood estimates of parameters in nonlinear mixed effect models, using a modern and efficient estimation algorithm, the stochastic approximation expectation maximisation (SAEM) algorithm.

Full details on `saemix` model setting & diagnostics can be found in "Parameter Estimation in Nonlinear Mixed Effect Models Using `saemix`, an R Implementation of the SAEM Algorithm" by Comets et al. 2017 (doi: 10.18637/jss.v080.i03)

Here we test the 2-steps VB curve (5 parameters) on *Solea solea* back-calculation data (30 fish GSA17 from Solemon Survey 2014-2020)

1. Load the dataset
2. Fit a Two step von Bertalanffy growth model:

$$\begin{aligned} 1) \quad & y(t) = \text{Lin}f(1 - \exp(-k(t - t_0))) \\ & \text{if } t < t_1 \\ 2) \quad & y(t) = \text{Lin}f(1 - \exp(-k_0(t_1 - t_0) - k_1(t - t_1))) \\ & \text{if } t > t_1 \end{aligned}$$

assuming the same population parameters for the two sex (sex can be introduced as covariate) using log-normal distributions for the parameters `Lin` and `t1`, logit for `k0` and `k1` and normal for `t0` assuming variance in `Lin`, `k0`, `t0`, `k1`, `t1` and assuming co-variance between all VB parameters.

Correlations between random effects can be introduced with the input argument `covariance.model`, a square matrix of size equal to the number of parameters in the model, giving the variance-covariance structure of the model: 1s correspond to estimated variances (in the diagonal) or covariances (off-diagonal elements).

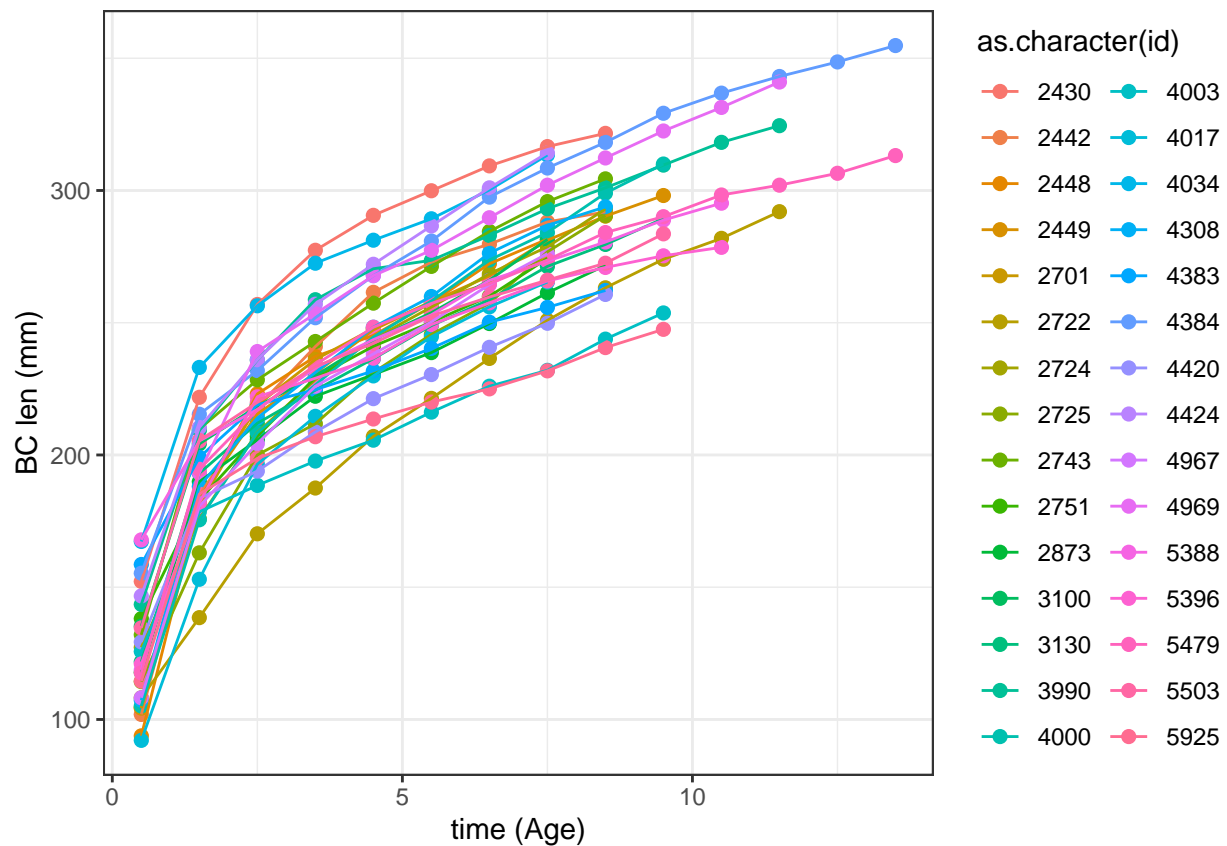
3. Check model diagnostic

Correlations plots Individual and population predictions Observations vs population predictions Scatter plot of residuals Residuals distribution Boxplots of the random effects Plots of the marginal distribution of the random effects Convergence plots Plotting Visual Predictiv Check

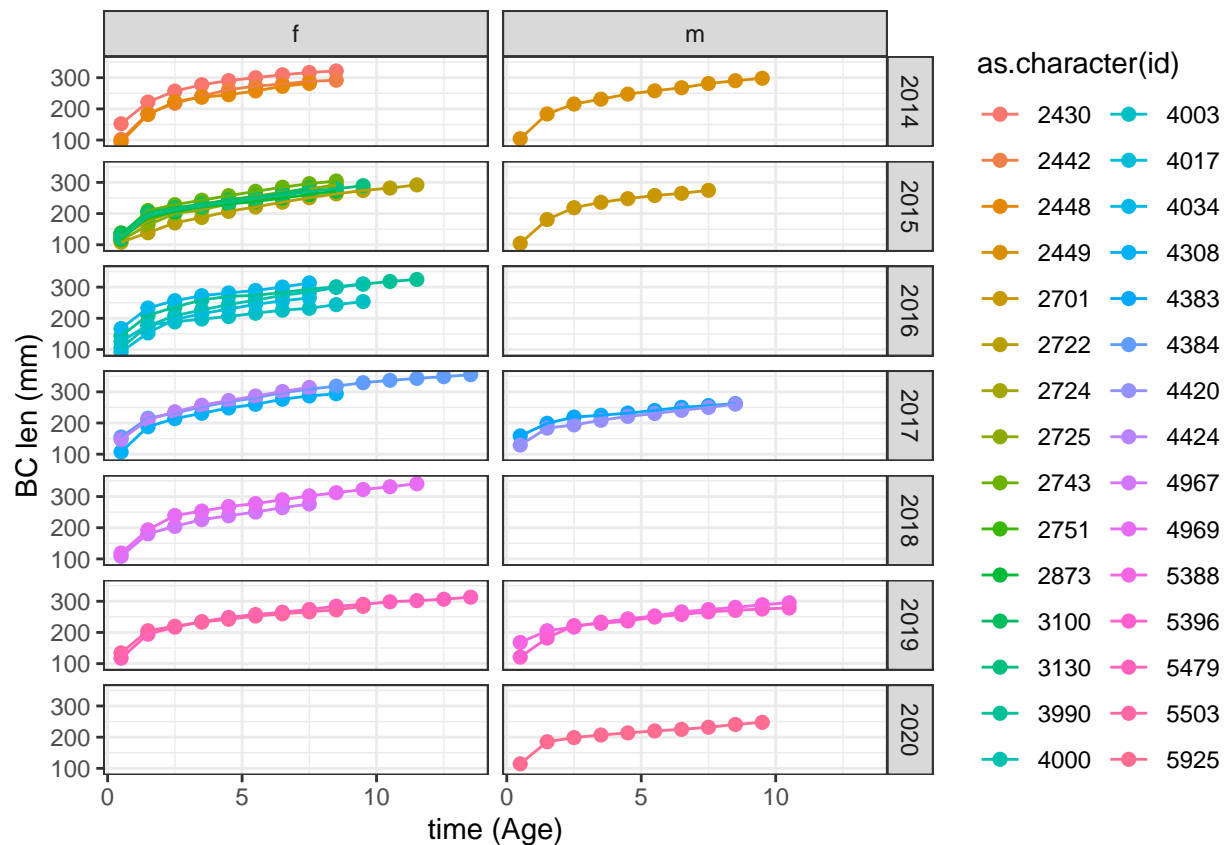
Data plotting

Plotting back-calculated length-at-age all together and by Sex & year

```
library(saemix)
library(FSA)
library(readr)
library(ggplot2)
library(dplyr)
setwd("C:/Users/f.masnadi/Desktop/Stock Assessment/SS3/SOLEA_SS3/dati/VBGP/Paper_bifasic/NonLinearMixEf
data_sol <- read.csv("data_saemixC.csv") %>% filter(Age > 0) %>% filter(id != "2740") %>% filter(id !=
ggplot(data=data_sol, aes(x=Age,y=TL, colour=as.character(id))) + geom_point( size=2) + xlab("time (Ag
```

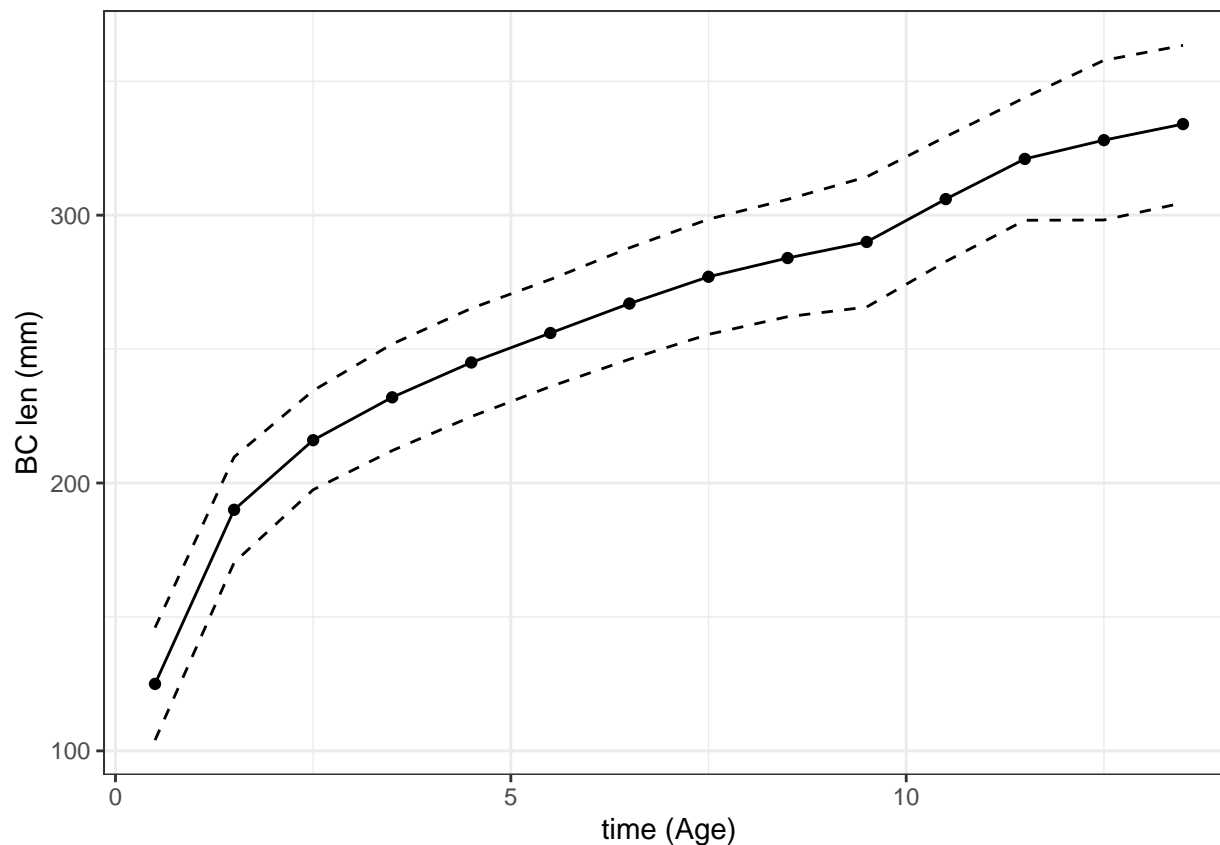


```
ggplot(data=data_sol, aes(x=Age,y=TL, colour=as.character(id))) + geom_point( size=2) + xlab("time (Ag
```



Calculate MEAN back-calculated length-at-age (+/- sd)

```
tmp <- data_sol %>%
  group_by(Age) %>%
  summarize(n=validn(TL),
            mn=round(mean(TL),0),
            sd=round(sd(TL),1)) %>%
  as.data.frame()
tmp$up <- tmp$mn + tmp$sd
tmp$dw <- tmp$mn - tmp$sd
ggplot(tmp, aes(Age, mn)) + geom_line() + geom_line(aes(Age, up), linetype = 2) + geom_line(aes(Age, dw))
```



Model setting

Create first the saemixData object

```
saemix.data<-saemixData(name.data=data_sol,
                        name.group=c("id"),
                        name.predictors=c("Age"),
                        # name.covariates = c("Sex"),
                        name.response=c("TL"))
```

```
## Using the object called data_sol in this R session as the data.
##
##
## The following SaemixData object was successfully created:
##
## Object of class SaemixData
##   longitudinal data for use with the SAEM algorithm
## Dataset data_sol
##   Structured data: TL ~ Age | id
##   Predictor: Age ()
```

Implement then the structural model and create the saemixModel object. Initial values for the population parameters should be provided.

```
vb.model <- function(psi,id,x) {
  Age <- x[,1]
  Linf<-psi[id,1]
  k0<-psi[id,2]
  t0<-psi[id,3]
  k1<-psi[id,4]
  t1<- psi[id,5]
  ypred <- (Age <= t1)*(Linf*(1-exp(-k0*(Age-t0)))) + (Age > t1)*(Linf*(1-exp(-k0*(t1-t0)-k1*(Age-t1)))
  return(ypred)
}
```

```
saemix.vb.model0<-saemixModel(model=vb.model,
  psi0=matrix(c(380,0.3,-0.5,0.2,1.8), ncol = 5, byrow = TRUE, dimnames = list(NULL, c("Linf", "k0",
  transform.par=c(1,1,0,1,1), #The distr for each parameter (0 = normal, 1 = log-normal, 2 =probit, 3 =
  fixed.estim=c(1,1,1,1,1), #Whether parameters should be estimated (1) or fixed to their initial es
  covariance.model = matrix(c(1, 1, 1, 1, 1,
                             1, 1, 1, 1, 1,
                             1, 1, 1, 1, 1,
                             1, 1, 1, 1, 1,
                             1, 1, 1, 1, 1), ncol = 5, byrow = TRUE))
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
## Model function
## function(psi,id,x) {
##   Age <- x[,1]
##   Linf<-psi[id,1]
##   k0<-psi[id,2]
##   t0<-psi[id,3]
##   k1<-psi[id,4]
##   t1<- psi[id,5]
##   ypred <- (Age <= t1)*(Linf*(1-exp(-k0*(Age-t0)))) + (Age > t1)*(Linf*(1-exp(-k0*(t1-t0)-k1*(Age-
##   return(ypred)
## }
## Nb of parameters: 5
## parameter names: Linf k0 t0 k1 t1
## distribution:
## Parameter Distribution Estimated
## [1,] Linf log-normal Estimated
## [2,] k0 log-normal Estimated
## [3,] t0 normal Estimated
## [4,] k1 log-normal Estimated
## [5,] t1 log-normal Estimated
## Variance-covariance matrix:
## Linf k0 t0 k1 t1
## Linf 1 1 1 1 1
## k0 1 1 1 1 1
## t0 1 1 1 1 1
## k1 1 1 1 1 1
## t1 1 1 1 1 1
```

```
## Error model: constant , initial values: a.=1
## No covariate in the model.
## Initial values
## Linf k0 t0 k1 t1
## Pop.CondInit 380 0.3 -0.5 0.2 1.8
```

#A square matrix of size equal to the number of parameters in the model, giving the variance-covariance

Run saemix for estimating the population parameters, computing the individual estimates, computing the FIM and the log-likelihood (linearization)

```
saemix.options<-saemixControl(map=TRUE, fim=TRUE, ll.is=FALSE, displayProgress=FALSE, seed=12345,nb.cha
saemix.vb.fit0 <- saemix(saemix.vb.model0,saemix.data,saemix.options)
```

Results

```
summary(saemix.vb.fit0)
```

```
## -----
## ----- Fixed effects -----
## -----
## Parameter Estimate      SE CV(%)
## 1      Linf    370.55 10.235 2.76
## 2        k0     0.35 0.032 8.95
## 3        t0    -0.72 0.096 13.36
## 4        k1     0.11 0.010 9.48
## 5        t1     1.53 0.114 7.44
## 6         a.     2.20 0.122 5.53
## -----
## ----- Variance of random effects -----
## -----
## Parameter Estimate      SE CV(%)
## Linf omega2.Linf    0.015 0.0056 38.18
## k0    omega2.k0     0.131 0.0546 41.63
## t0    omega2.t0     0.156 0.0605 38.90
## k1    omega2.k1     0.190 0.0681 35.88
## t1    omega2.t1     0.102 0.0378 36.96
## -----
## ----- Correlation matrix of random effects -----
## -----
## omega2.Linf omega2.k0 omega2.t0 omega2.k1 omega2.t1
## omega2.Linf 1.00      -0.64 -0.41 -0.74 0.08
## omega2.k0   -0.64      1.00 0.81 0.44 -0.61
## omega2.t0   -0.41      0.81 1.00 0.29 -0.40
## omega2.k1   -0.74      0.44 0.29 1.00 0.10
## omega2.t1   0.08      -0.61 -0.40 0.10 1.00
## -----
## ----- Statistical criteria -----
## -----
## Likelihood computed by linearisation
```

```
##      -2LL= 1789.401
##      AIC = 1831.401
##      BIC = 1860.827
## -----
```

For each parameter estimated in the model, estimates of the standard error are reported, as an absolute value (SE) and relative to the estimate, as a coefficient of variation (% CV). In the present case, the fixed parameters Linf, k0, t1, k1 are well estimated, with coefficients of variation about 10%. t0 have a greater variation (11%). CV for random effects are quite high (>30%) maybe reflecting the limited information available with only 30 subjects in the dataset (119 in Alos et al. 2010).

Individual parameters value:

```
psi <- psi(saemix.vb.fit0)
psi
```

##		Linf	k0	t0	k1	t1
## 1	344.5620	0.4526846	-0.7855132	0.22695028	2.0096909	
## 2	307.3241	0.5146648	-0.2837199	0.30173935	1.6285393	
## 3	382.2924	0.3629832	-0.2791769	0.09087553	1.9799350	
## 4	370.7066	0.3524961	-0.4410060	0.10844650	1.8165863	
## 5	321.2955	0.4348756	-0.4061870	0.15526318	2.1007068	
## 6	389.4964	0.1284563	-1.9662621	0.09052285	2.5158034	
## 7	386.4622	0.2152337	-1.3398925	0.10270837	2.1091304	
## 8	455.0362	0.1550403	-1.3628128	0.07195706	2.1106232	
## 9	398.4656	0.3433118	-0.6733557	0.09996495	1.5000000	
## 10	355.4189	0.2467411	-1.4616044	0.11312091	1.8674897	
## 11	386.3664	0.3121140	-0.6670505	0.07228529	1.5641815	
## 12	417.8066	0.2813263	-0.8954768	0.07114207	1.4564148	
## 13	398.7877	0.2964588	-0.7330014	0.07601600	1.5666171	
## 14	389.4954	0.3146046	-0.9567405	0.09015588	1.9330164	
## 15	412.2975	0.2589651	-0.6405496	0.09662298	1.8171382	
## 16	414.1073	0.3019008	-0.6966128	0.04654243	1.0863920	
## 17	339.7948	0.2832622	-0.6193689	0.13308343	2.3750385	
## 18	414.9026	0.3143894	-1.1339200	0.08407299	1.7274002	
## 19	360.8502	0.3819677	-0.4292043	0.13388818	1.5826986	
## 20	340.2651	0.4928388	-0.7686237	0.08224492	0.9866652	
## 21	405.4158	0.3256024	-0.9771566	0.11236460	1.2276971	
## 22	379.7859	0.3707023	-0.6213099	0.07059867	1.0618770	
## 23	408.0468	0.2821391	-1.0731222	0.11805805	1.6391644	
## 24	363.0776	0.3297953	-0.5812565	0.11649258	1.6532081	
## 25	449.1264	0.2546709	-0.7090023	0.07267844	2.1590368	
## 26	356.8744	0.4546429	-0.3499878	0.05357561	1.2593495	
## 27	369.8031	0.4633715	-0.7973879	0.08847471	0.8186147	
## 28	325.0733	0.3606006	-0.7873586	0.10877695	2.2494804	
## 29	340.2551	0.4234288	-0.5029988	0.13324647	1.5795108	
## 30	371.5606	0.3543406	-0.7708783	0.07420122	1.5900466	

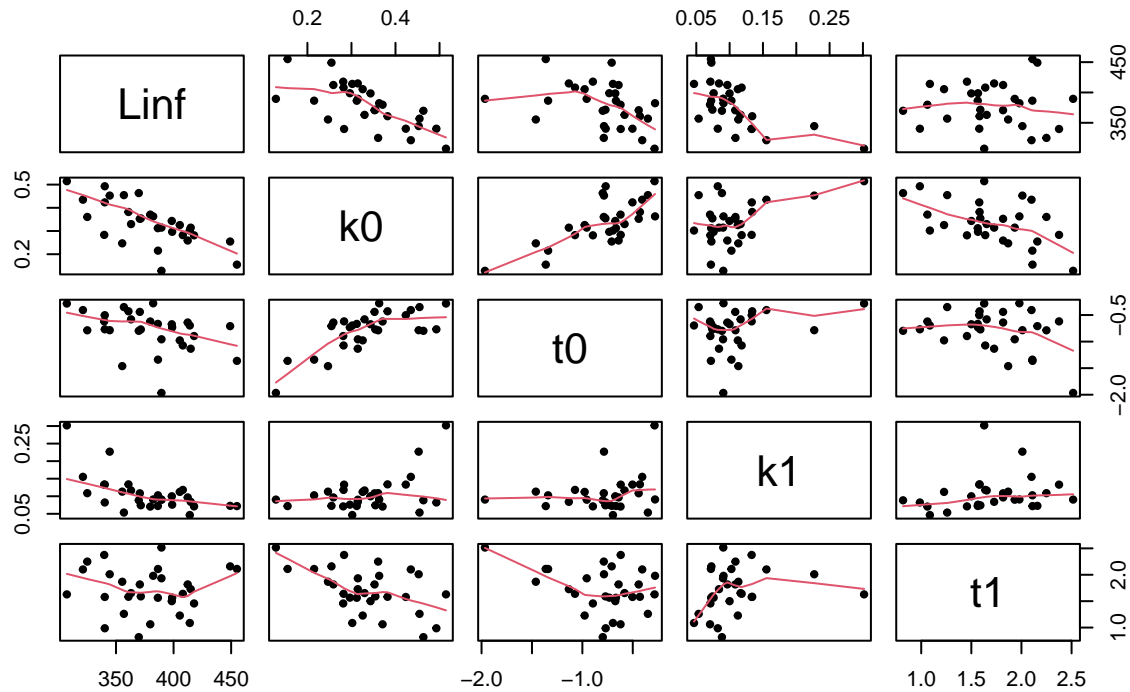
Diagnostic

Display some diagnostic plots:

Correlations plots

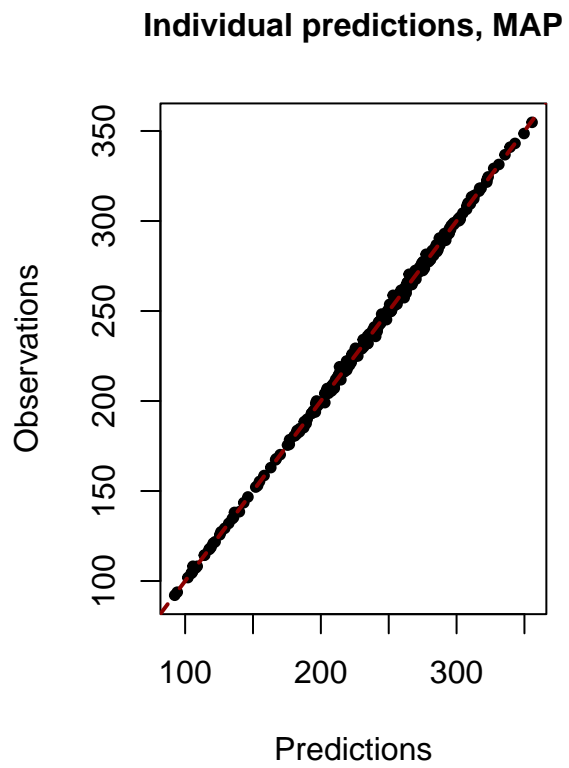
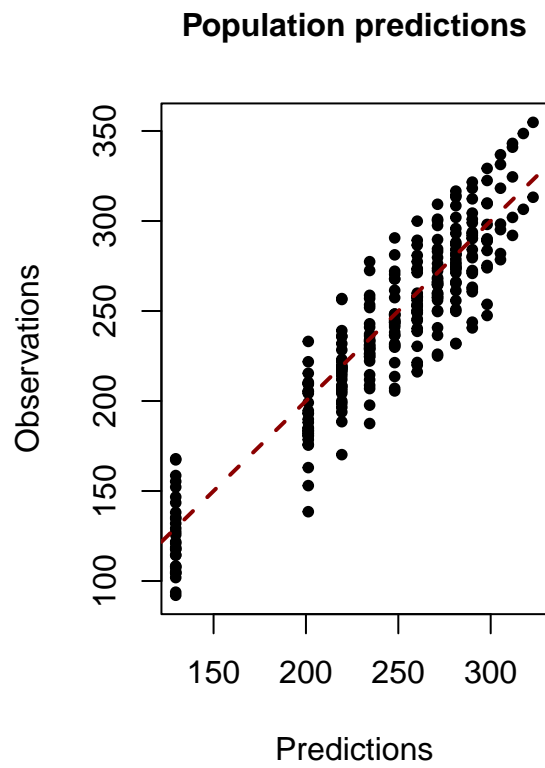
```
saemix.plot.select(saemix.vb.fit0, correlations = T)
```

Correlations between random effects



Observations vs population predictions In Bayesian statistics, Maximum A Posteriori estimate (MAP) is an estimate of an unknown quantity, that equals the mode of the posterior distribution.

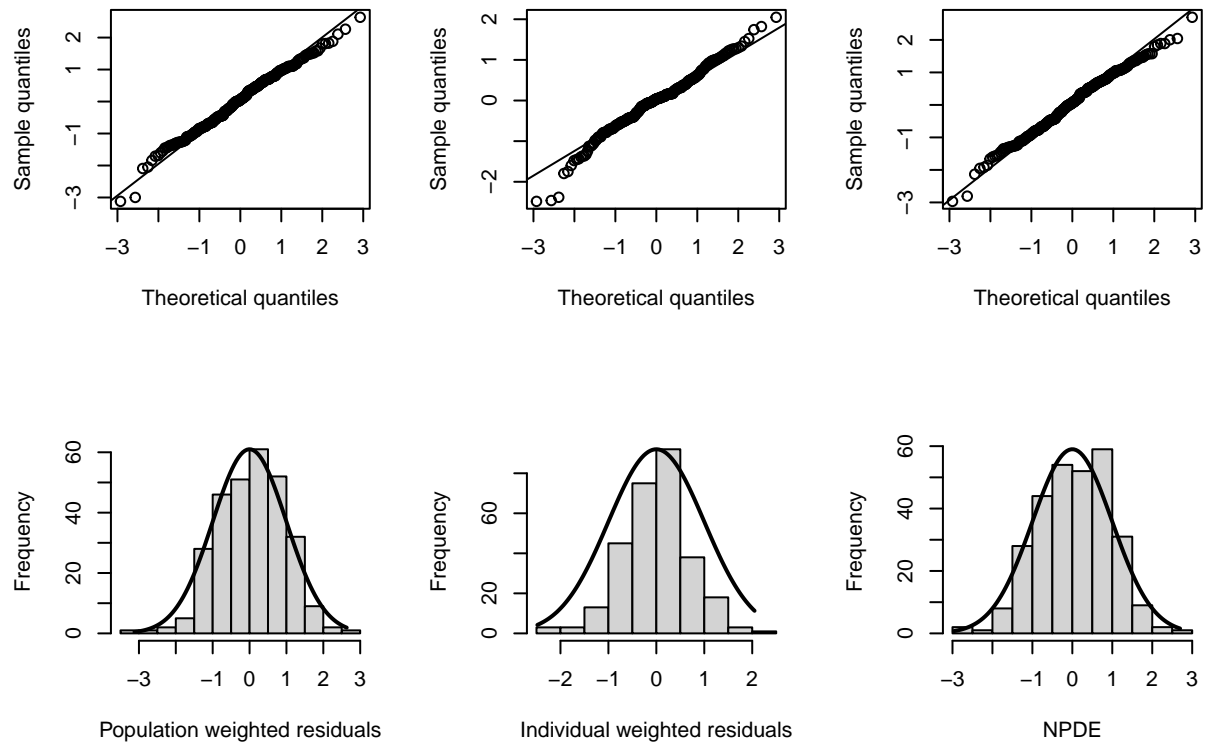
```
saemix.plot.obsvspred(saemix.vb.fit0)
```

Residuals distribution

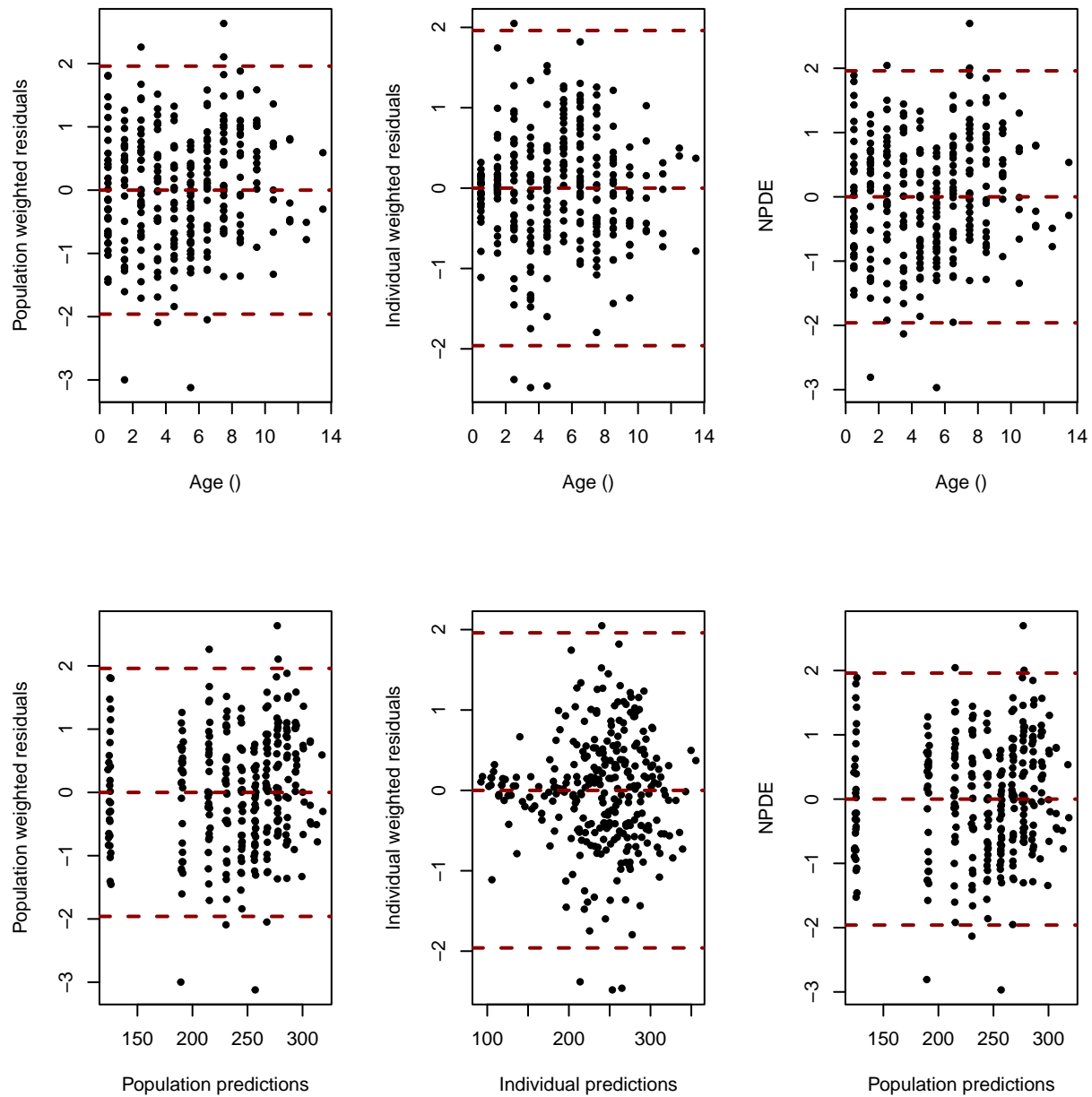
```
saemix.plot.select(saemix.vb.fit0,residuals.distribution = T)
```

```
## Plotting the distribution of residuals
```



Scatter plot of residuals NPDE : Normalised Prediction Distribution Errors (Comets E, Brendel K, Mentré F (2008).)

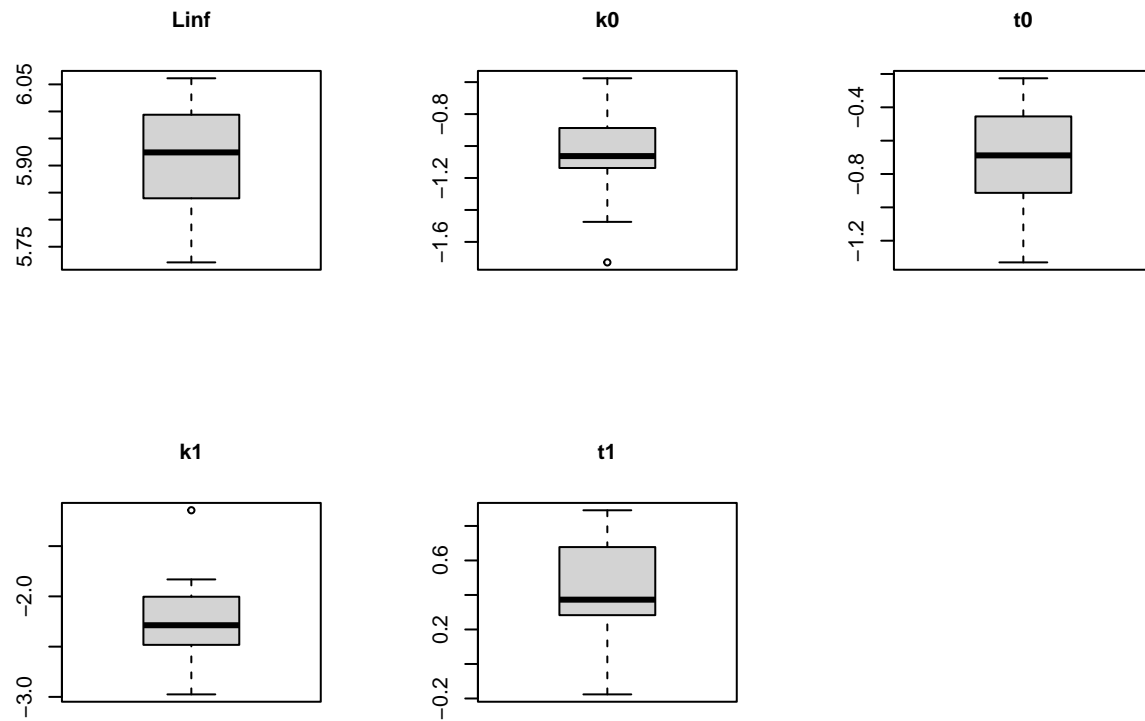
```
saemix.plot.scatterresiduals(saemix.vb.fit0 )
```



Residuals distribution

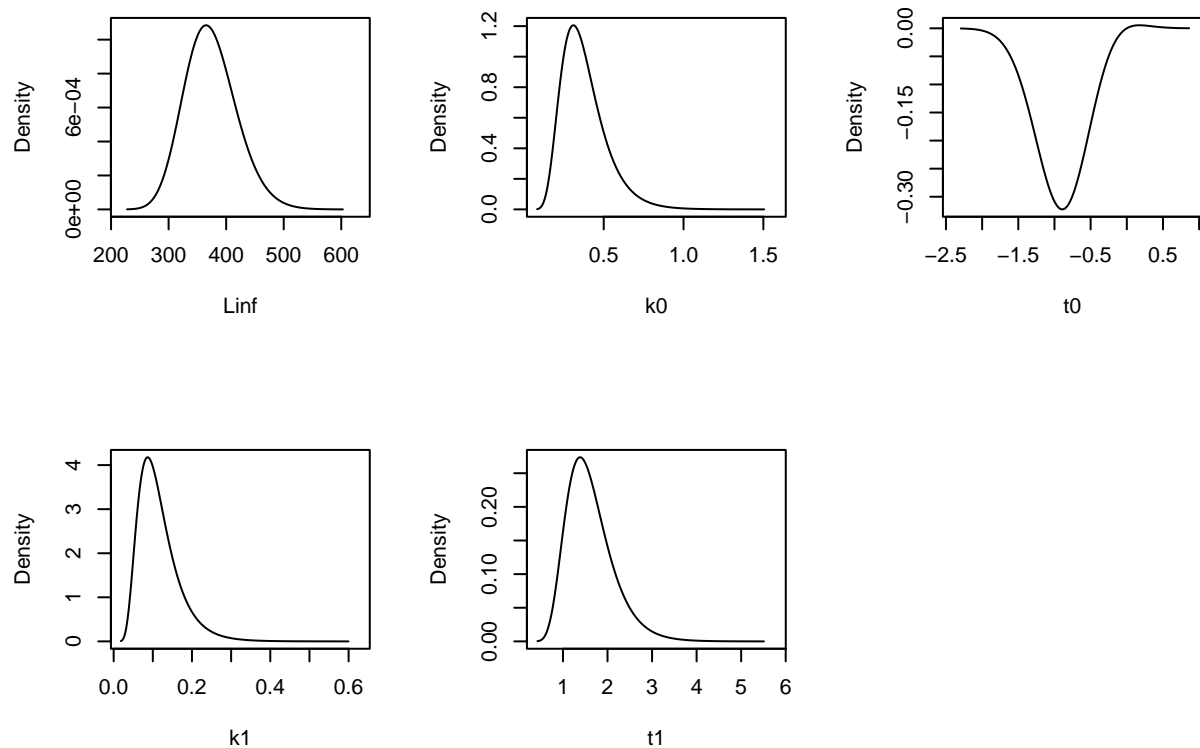
Boxplots of the random effects

```
saemix.plot.select(saemix.vb.fit0, random.effects = T)
```



Plots of the marginal distribution of the random effects. It gives the probabilities of various values of the variables in the subset without reference to the values of the other variables.

```
saemix.plot.select(saemix.vb.fit0, marginal.distribution = T)
```



```
# Convergence plots
# saemix.plot.select(saemix.vb.fit0,convergence =T)
```

Classic (3-par) VB

The same dataset was used to fit a classical VB curve (only 3 parameters: Linf, k, t0)

Results Classic VB

```
summary(saemix.vb.fit3)
```

```
## -----
## ----- Fixed effects -----
## -----
##   Parameter Estimate    SE CV(%)
## 1      Linf      295.47 5.446  1.84
## 2         k       0.33 0.021  6.53
## 3        t0      -1.42 0.131  9.21
## 4         a       8.22 0.408  4.97
## -----
## ----- Variance of random effects -----
```

```
## -----
##           Parameter Estimate      SE CV(%)
## Linf omega2.Linf  0.0089 0.0026 29.35
## k      omega2.k   0.0962 0.0330 34.28
## t0     omega2.t0  0.3736 0.1312 35.13
## -----
## ----- Correlation matrix of random effects -----
## -----
##           omega2.Linf omega2.k omega2.t0
## omega2.Linf  1.00      -0.61  -0.36
## omega2.k     -0.61      1.00   0.81
## omega2.t0    -0.36      0.81   1.00
## -----
## ----- Statistical criteria -----
## -----
## Likelihood computed by linearisation
##      -2LL= 2266.598
##      AIC = 2286.598
##      BIC = 2300.61
## -----
```

Individual parameters value:

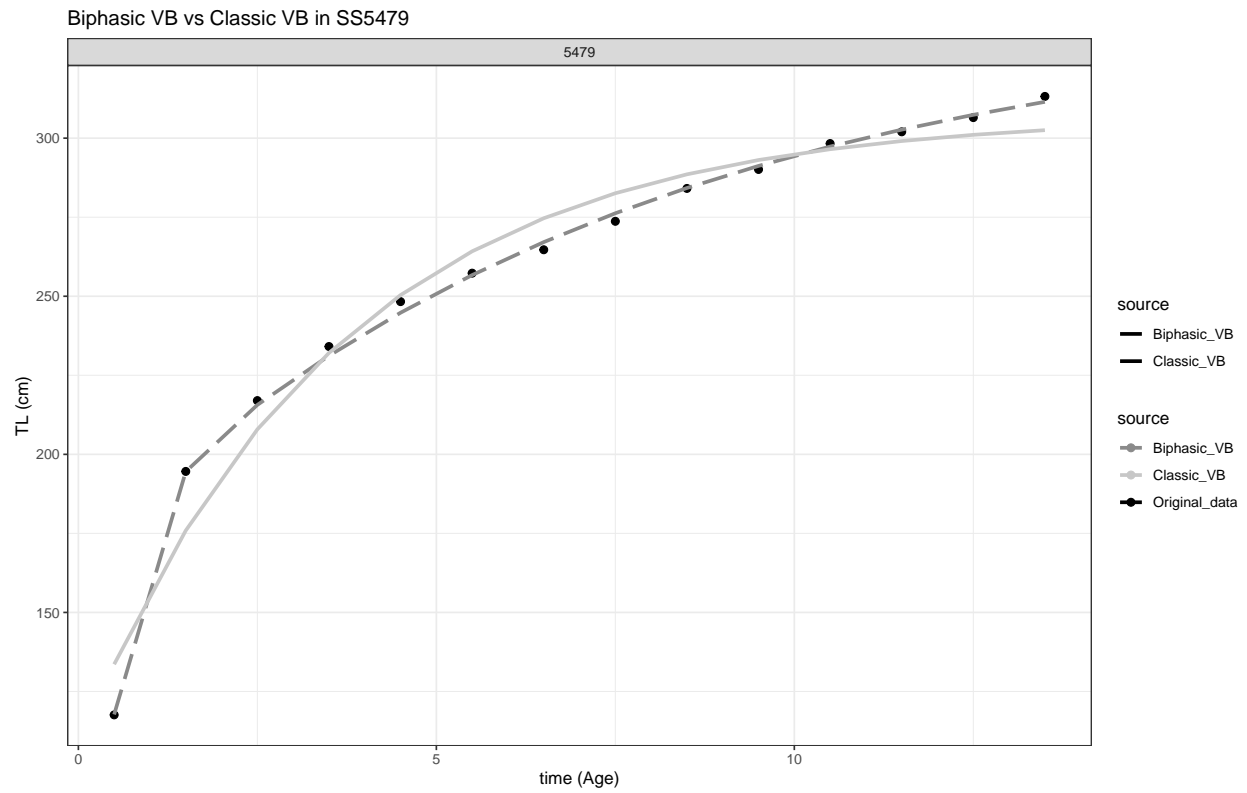
```
psi3 <- psi(saemix.vb.fit3)
psi3
```

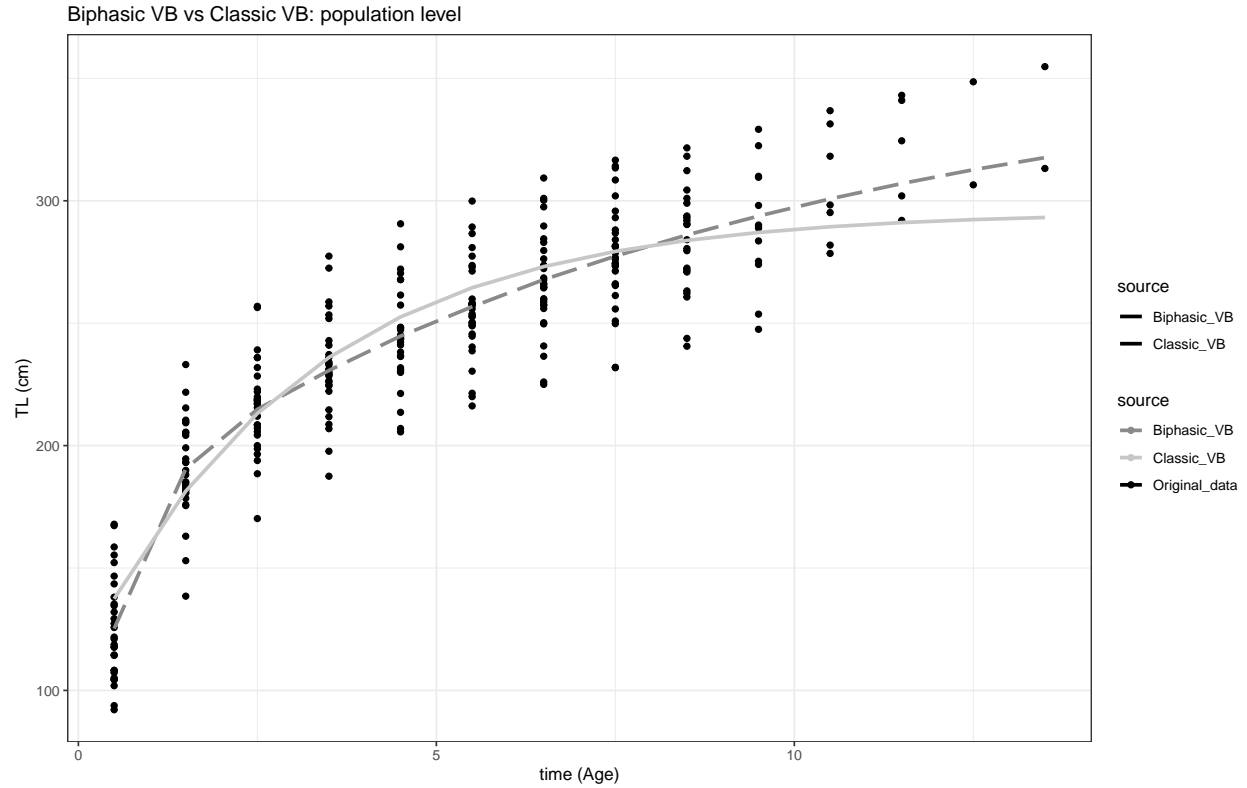
```
##           Linf           k           t0
## 1  324.2686 0.4170377 -1.0868884
## 2  294.2000 0.4432482 -0.5430147
## 3  277.1543 0.5268687 -0.3593801
## 4  295.3161 0.3630233 -0.8706637
## 5  274.6141 0.5005616 -0.5230157
## 6  326.9473 0.1522493 -2.2206697
## 7  303.1772 0.2812309 -1.5094199
## 8  308.6699 0.2384823 -1.5763115
## 9  306.8738 0.3452886 -1.3084590
## 10 284.9652 0.3165395 -1.6100908
## 11 268.6179 0.3861929 -1.1679964
## 12 282.9476 0.3671247 -1.4180220
## 13 289.1093 0.3141275 -1.4641368
## 14 318.4622 0.3129506 -1.5734651
## 15 315.3598 0.2791484 -1.1553656
## 16 253.1492 0.2932931 -1.9699827
## 17 272.6106 0.3890050 -0.6258822
## 18 313.6624 0.4149354 -1.4058540
## 19 296.3750 0.3654352 -0.8935959
## 20 261.9076 0.3960060 -1.7730578
## 21 361.6704 0.2029422 -2.3912298
## 22 265.2258 0.3087887 -1.7722131
## 23 324.1849 0.3264151 -1.4316554
## 24 280.1465 0.3834885 -0.9031511
## 25 335.1929 0.2923328 -1.2203585
## 26 238.8389 0.4882361 -0.9812584
```

```
## 27 296.4141 0.2713874 -2.3811189
## 28 276.1180 0.3962633 -1.0695385
## 29 307.1109 0.2794686 -1.5431600
## 30 277.2363 0.3958489 -1.3333896
```

Comparison Classic (3-par) vs Biphasic (5-par) VB

Visual comparison on SS5479

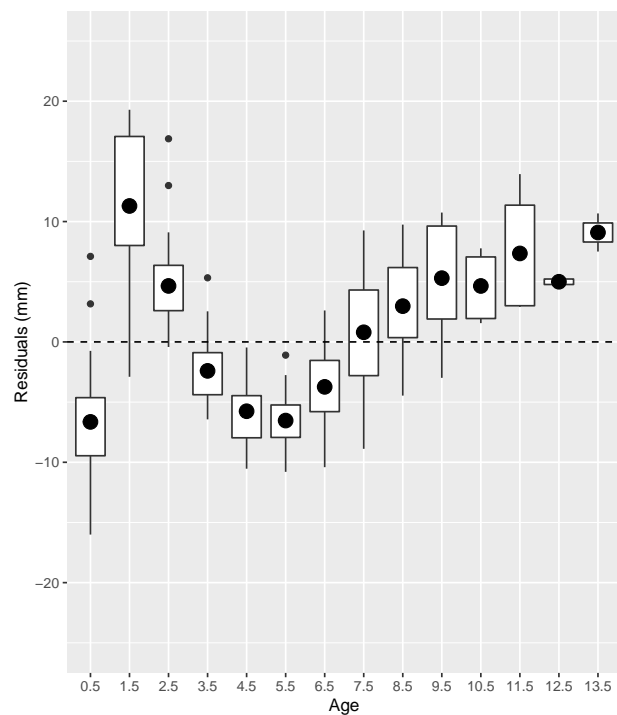




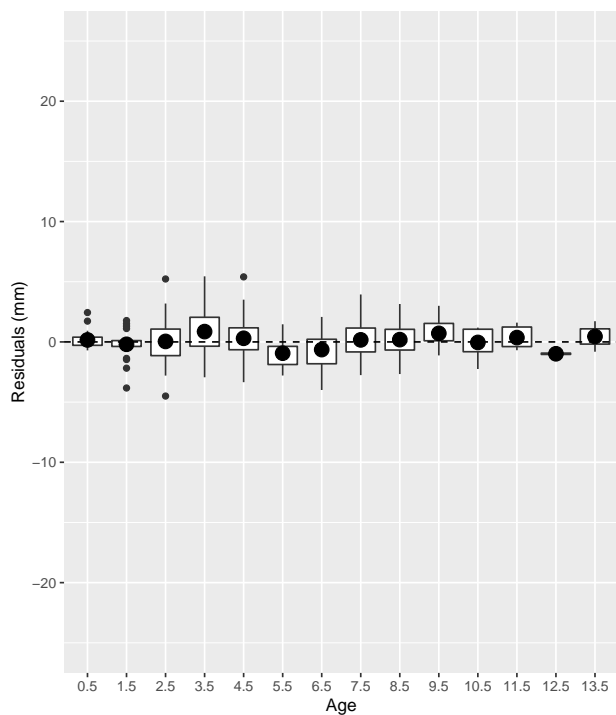
Residuals boxplots

Box plots of the residuals of the von Bertalanffy growth models based on three and five parameters, defined as the observed length-at-age minus the predicted posterior mean of the model.

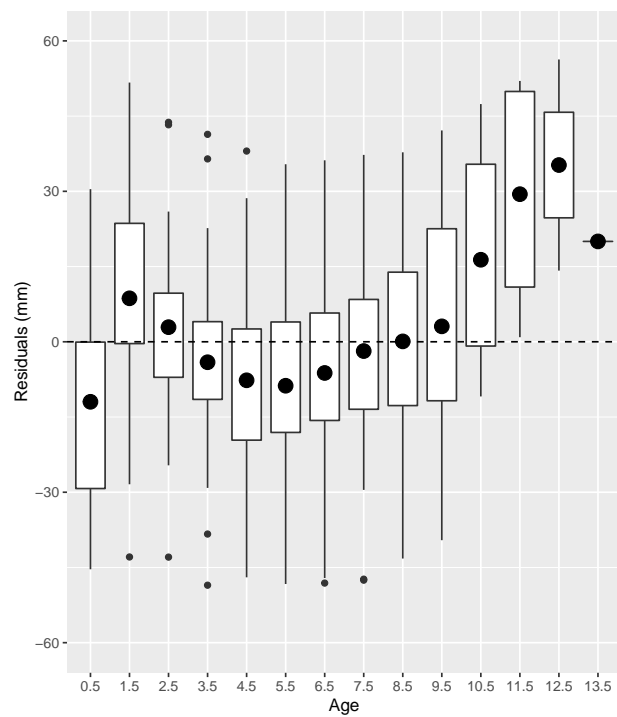
3-par VB residuals (mm) by Age
Individual prediction



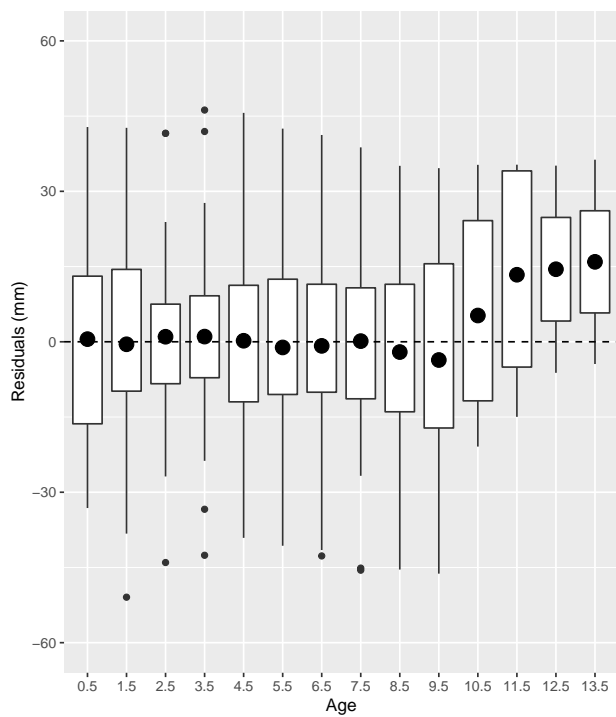
5-par VB residuals (mm) by Age
Individual prediction



3-par VB residuals (mm) by Age
Population prediction

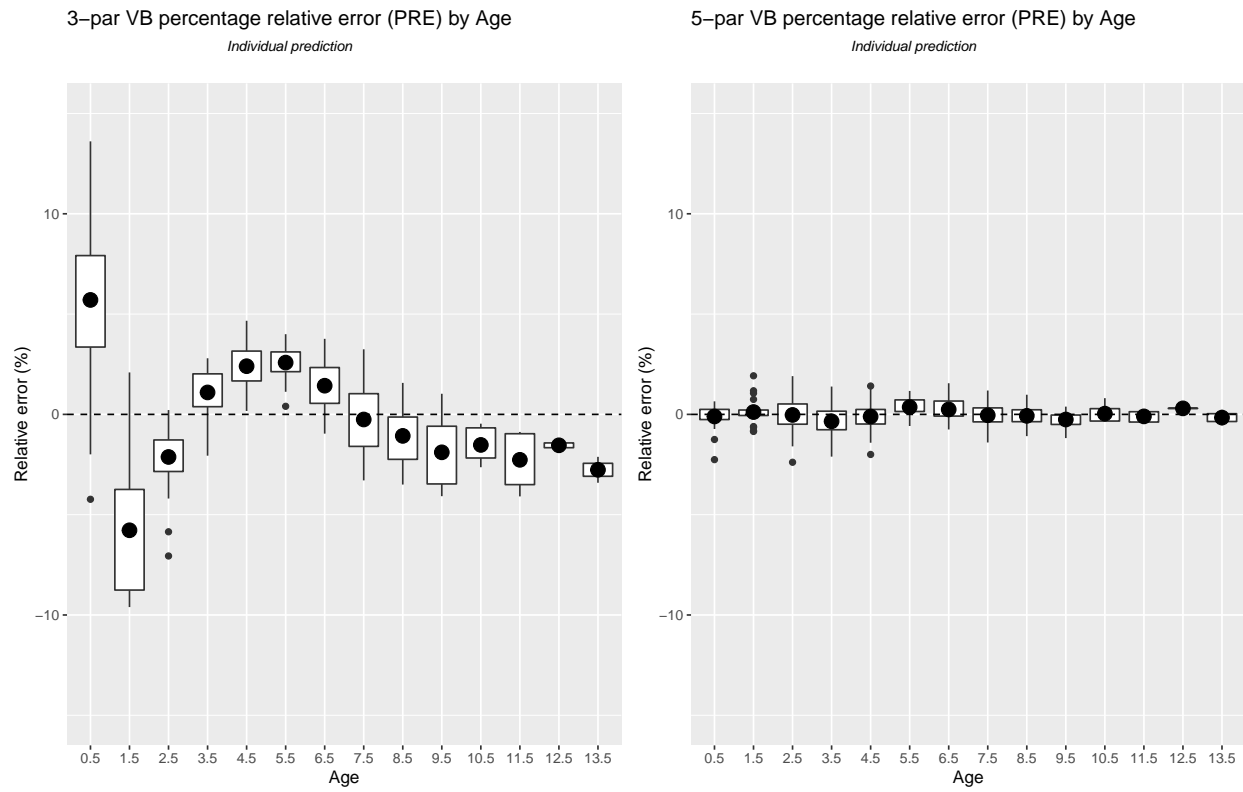


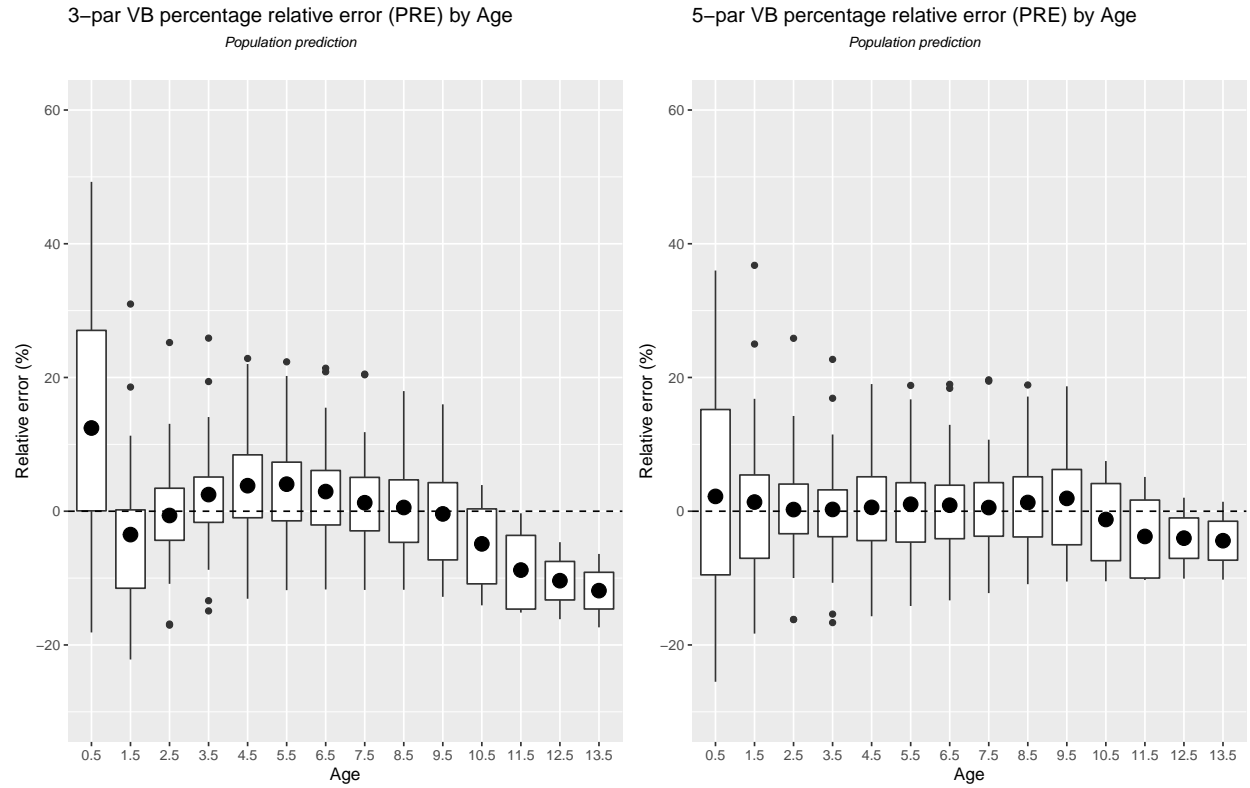
5-par VB residuals (mm) by Age
Population prediction



Percentage relative error (PRE) boxplots

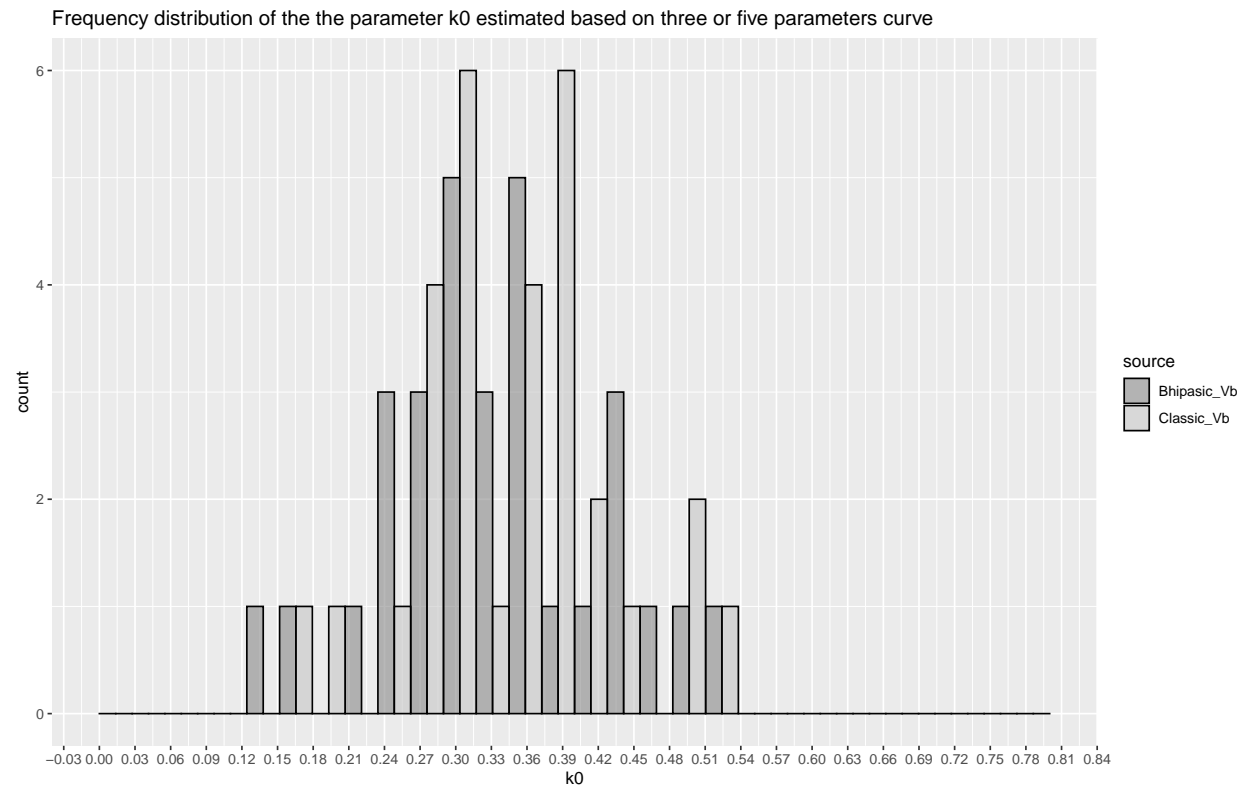
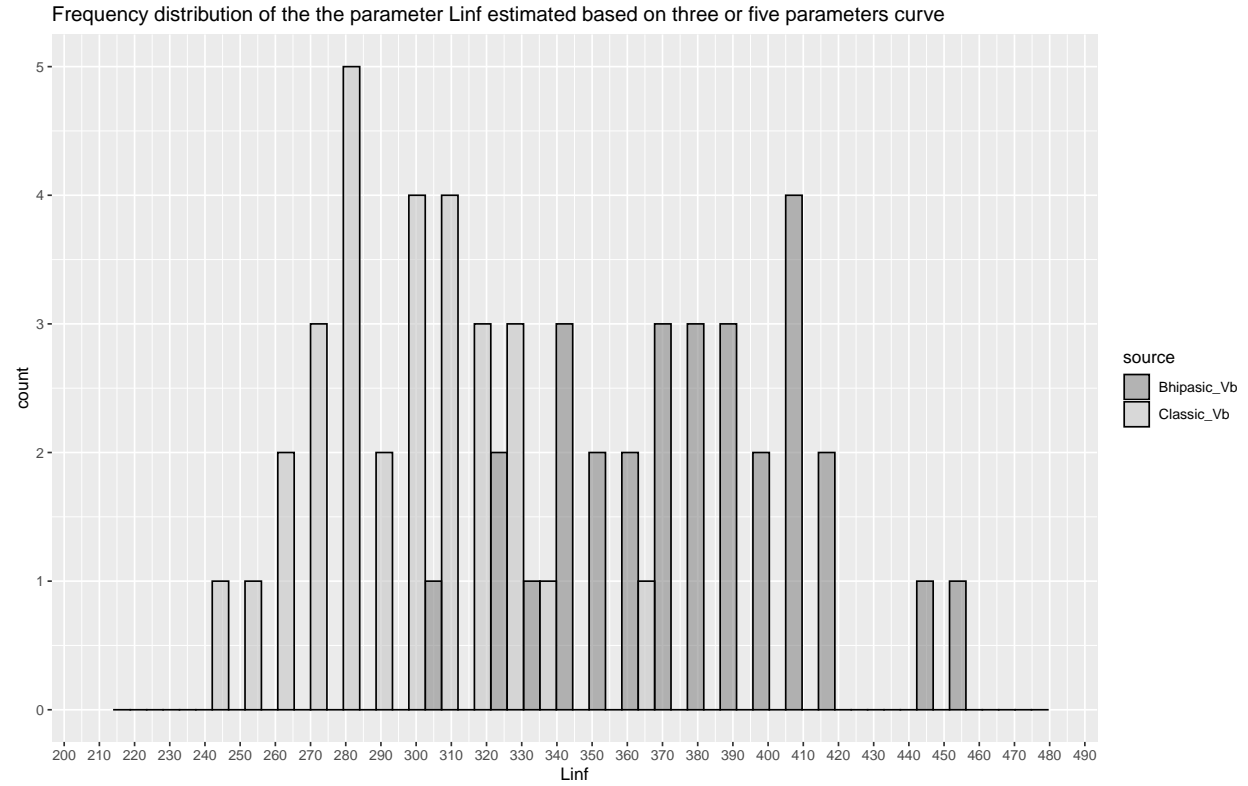
The relative error is the absolute error divided by the magnitude of the exact value. The percent relative error is the relative error expressed in terms of per 100.

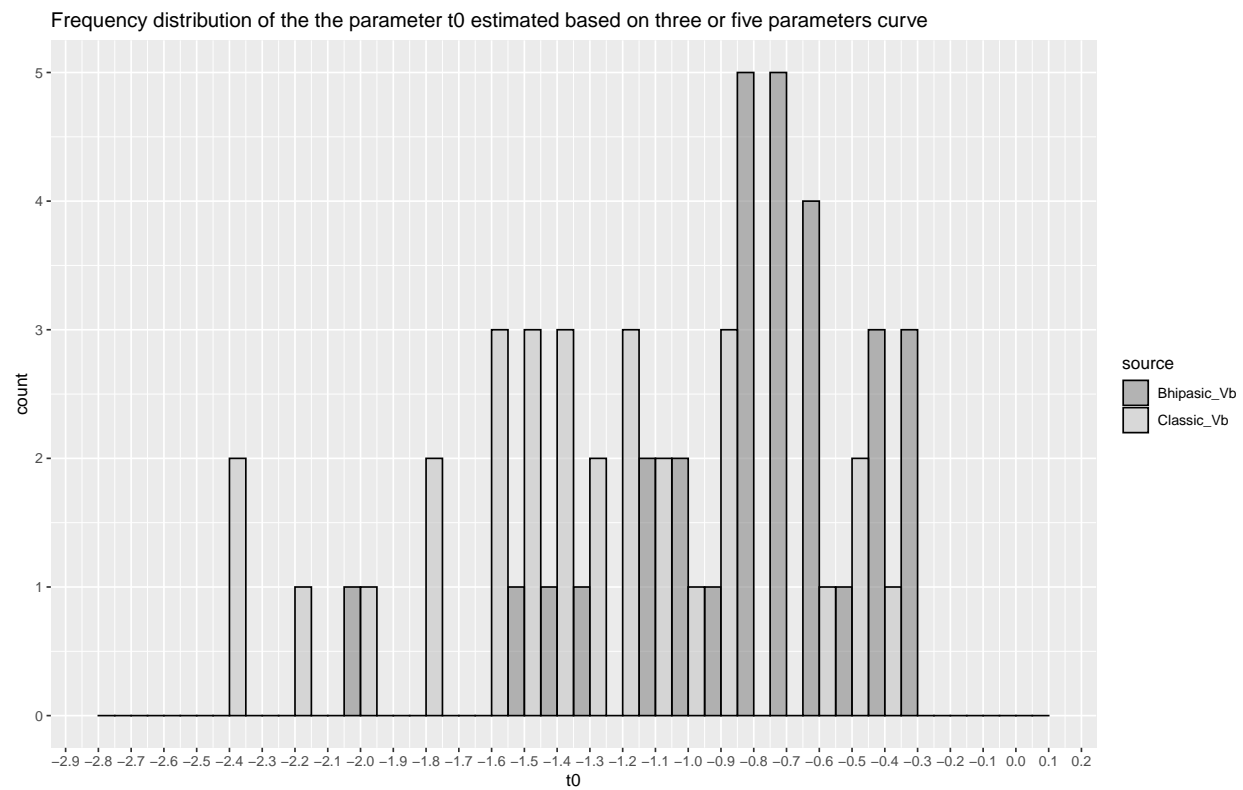
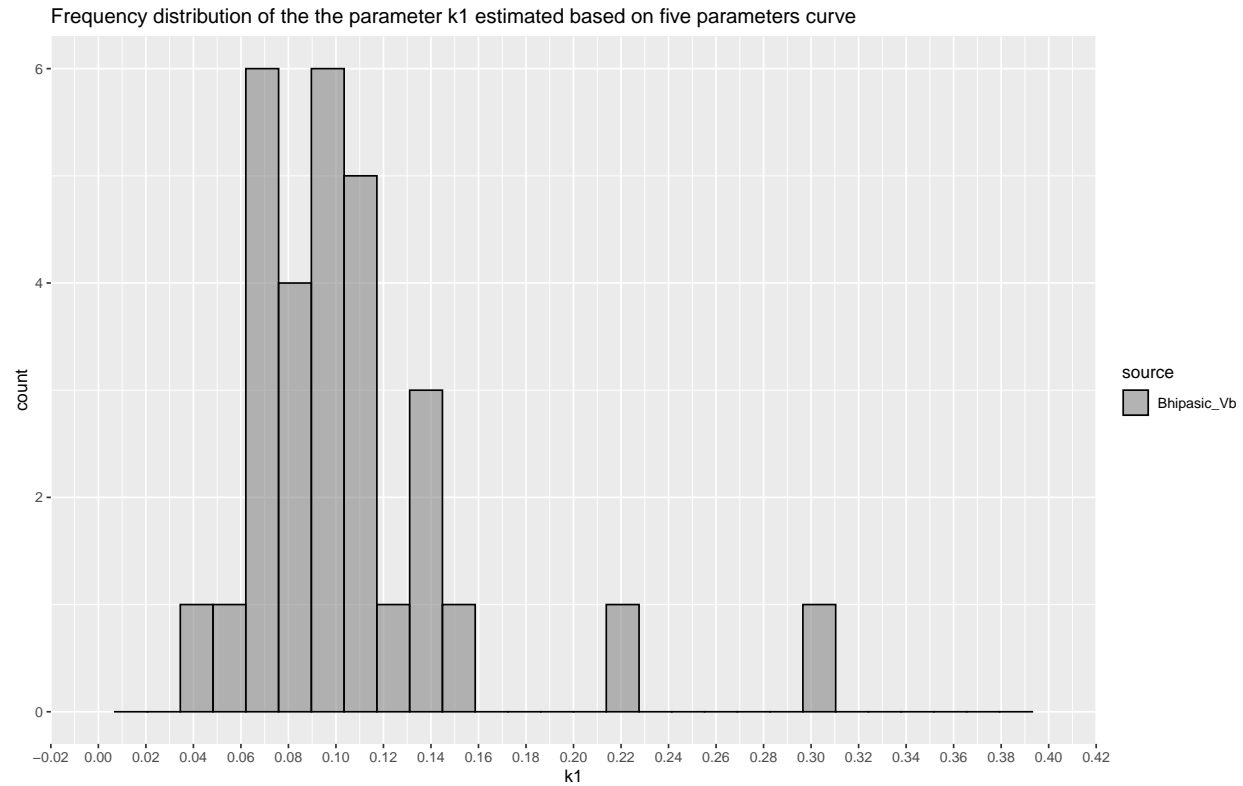


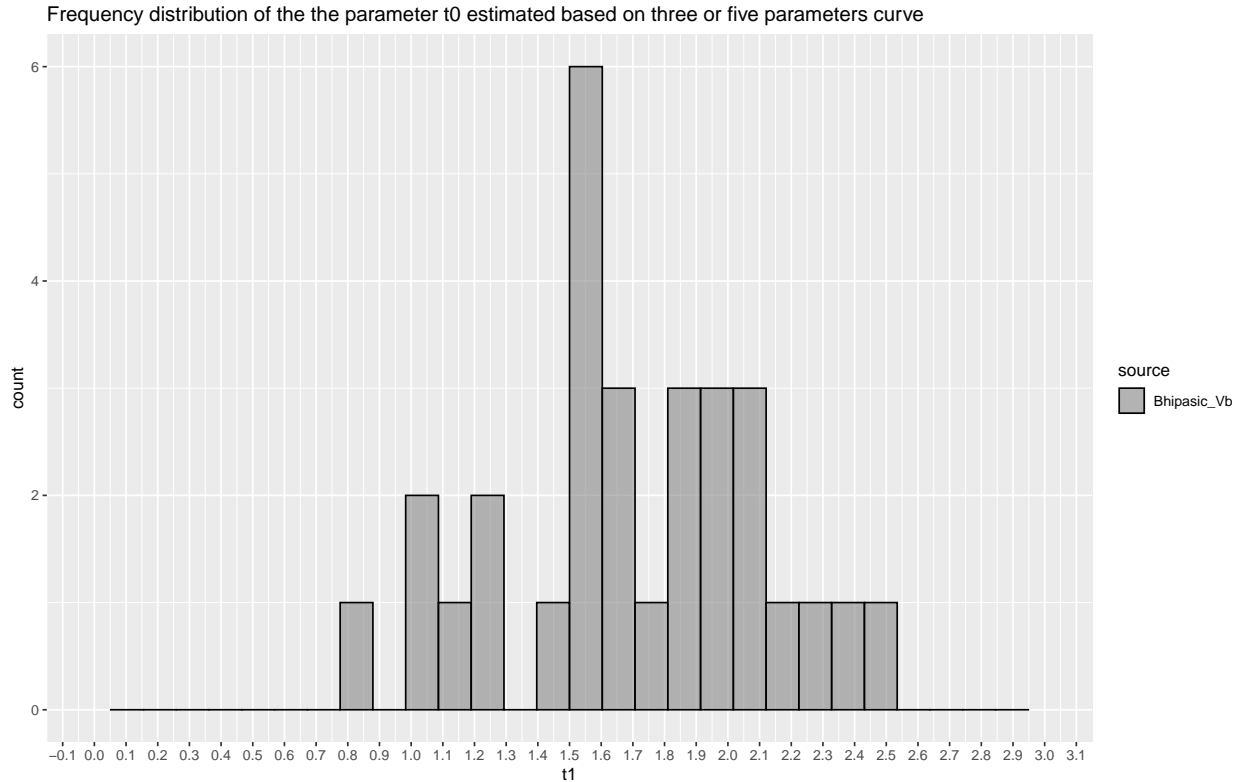


Frequency distribution of “in common” individual parameters Linf and t_0

Linf is higher for the biphasic curve and t_0 is closer to 0 respect to the classic VB







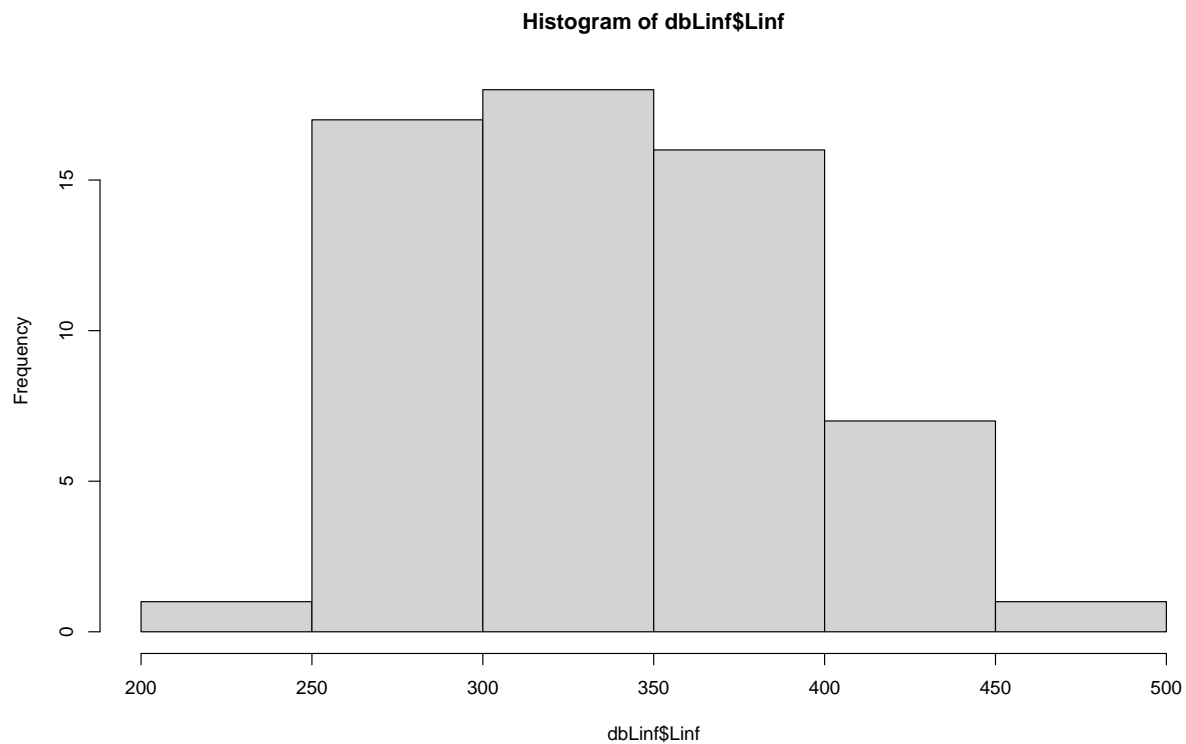
```
## # A tibble: 2 x 4
##   source      min    max   Med
##   <chr>      <dbl> <dbl> <dbl>
## 1 Bhipasic_Vb 307.  455. 381.
## 2 Classic_Vb  239.  362. 295.
```

```
## # A tibble: 2 x 4
##   source      min    max   Med
##   <chr>      <dbl> <dbl> <dbl>
## 1 Bhipasic_Vb 0.128 0.515 0.328
## 2 Classic_Vb  0.152 0.527 0.354
```

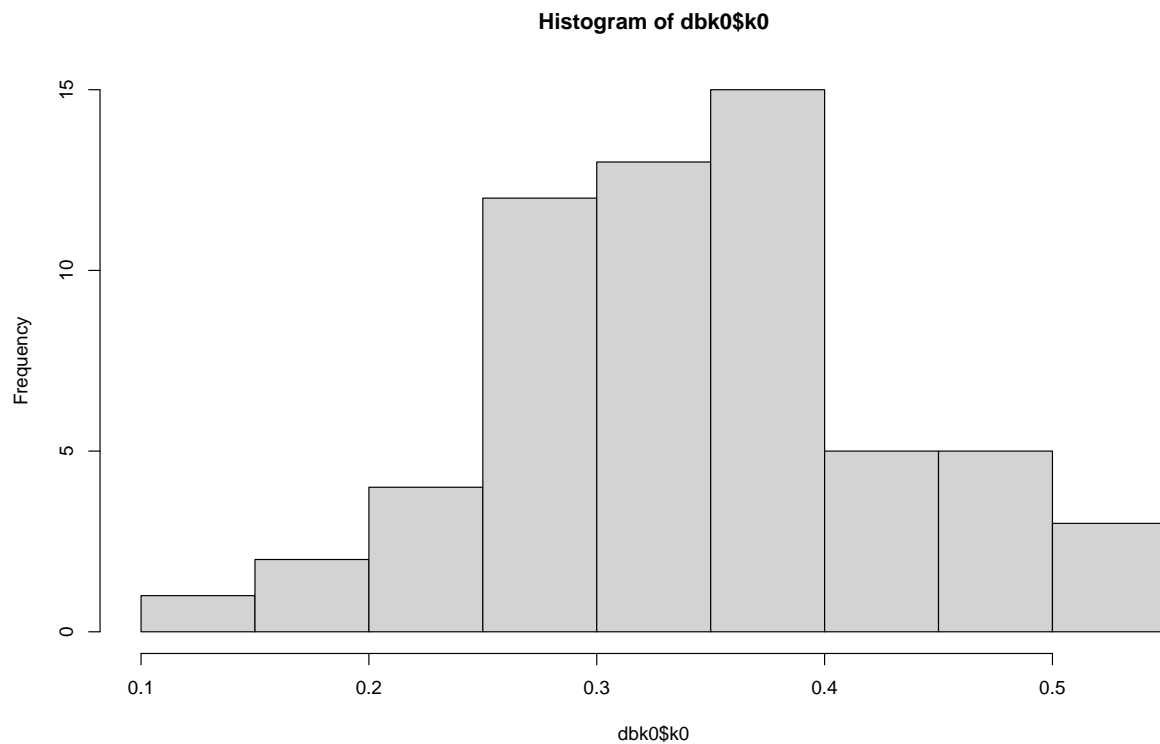
```
## # A tibble: 2 x 4
##   source      min    max   Med
##   <chr>      <dbl> <dbl> <dbl>
## 1 Bhipasic_Vb -1.97 -0.279 -0.721
## 2 Classic_Vb  -2.39 -0.359 -1.37
```

```
## # A tibble: 1 x 4
##   source      min    max   Med
##   <chr>      <dbl> <dbl> <dbl>
## 1 Bhipasic_Vb 0.819  2.52  1.65
```

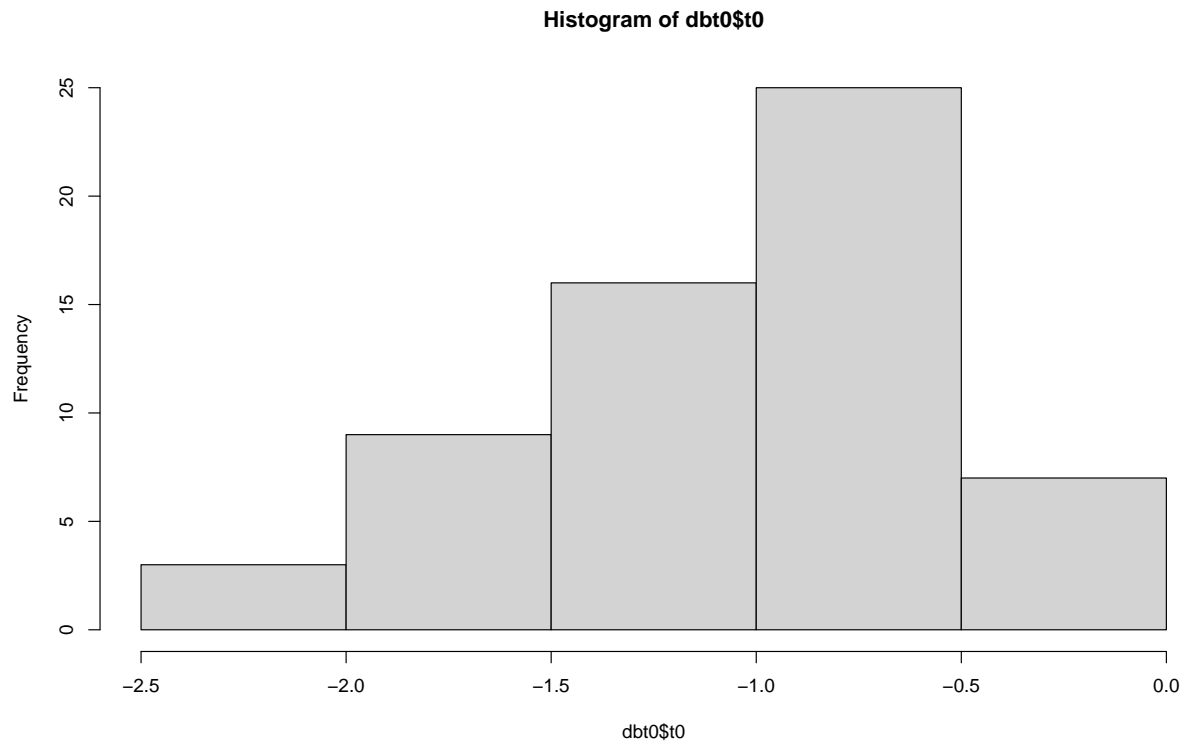
```
## # A tibble: 1 x 4
##   source      min    max   Med
##   <chr>      <dbl> <dbl> <dbl>
## 1 Bhipasic_Vb 0.0465 0.302 0.0937
```



```
##
## Welch Two Sample t-test
##
## data: Linf by source
## t = 10.301, df = 53.522, p-value = 2.63e-14
## alternative hypothesis: true difference in means between group Bhipasic_Vb and group Classic_Vb is not equal to 0
## 95 percent confidence interval:
##  67.76117 100.52055
## sample estimates:
## mean in group Bhipasic_Vb mean in group Classic_Vb
##          378.4918          294.3509
```



```
##
## Welch Two Sample t-test
##
## data: k0 by source
## t = -0.56006, df = 57.576, p-value = 0.5776
## alternative hypothesis: true difference in means between group Bhipasic_Vb and group Classic_Vb is not equal to 0
## 95 percent confidence interval:
## -0.05860357 0.03298286
## sample estimates:
## mean in group Bhipasic_Vb mean in group Classic_Vb
## 0.3354537 0.3482640
```

```
##
## Welch Two Sample t-test
##
## data:  t0 by source
## t = 4.6752, df = 52.975, p-value = 2.061e-05
## alternative hypothesis: true difference in means between group Bhipasic_Vb and group Classic_Vb is not equal to 0
## 95 percent confidence interval:
##  0.3115977 0.7798561
## sample estimates:
## mean in group Bhipasic_Vb mean in group Classic_Vb
## -0.7903513 -1.3360782
```

Final conclusion

Models can be compared through the Akaike criterion (AIC) and Schwarz's information criterion, also called Bayes information criterion or BIC.

Here a the comparison between the Classic (3-par) vs Biphasic (5-par) VB for these 32 fishes. Model selection via statistical criteria selected the Biphasic model as the best one (AIC and BIC lower than the Classic one). 7/8 cm difference in Linf.

```
## # A tibble: 2 x 8
##   Model   'L8 (CV%)' 'k0 (CV%)' 't0 (CV%)' 'k1 (CV%)' 't1 (CV%)' AIC BIC
##   <chr>   <chr>      <chr>      <chr>      <chr>      <chr>      <dbl> <dbl>
## 1 3-par VB 295 (1.8) 0.33 (6.3) -1.41 (9.1) NA         NA         2278 2292
## 2 5-par VB 375 (3.2) 0.34 (8.9) -0.76 (12.7) 0.10 (10) 1.56 (6.8) 1810 1839
```

On an individual level, the biphasic curve fits the data much better than the classic VB curve (younger and older ages).

At the population level, the residues are quite high for both models, however the trend observed at the individual level is replicated.

Despite the paucity of data, we can already see the growth change around 2 years ($t_1=1.6$) with k ($k_0=0.34$; $k_1=0.11$) decreasing after this age. Thanks to this variation of the k rate between k_0 and k_1 , L_{inf} is higher in the biphasic curve compared to the classic one (37 cm vs 29 cm) providing a better matching with the L_{max} data for common sole (39 cm). Age > 10 are estimated worse because only few records in the data.

In conclusion, individual fitting using biphasic pattern is better than classic one but the variability given by the few samples (32 fish) could be maybe too high to have acceptable population level results. Nevertheless, the advantages of the biphasic curve remain proven because the high variability at the population level is shared in both models. #